PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-251289
(43)Date of publication of application : 14.09.2001

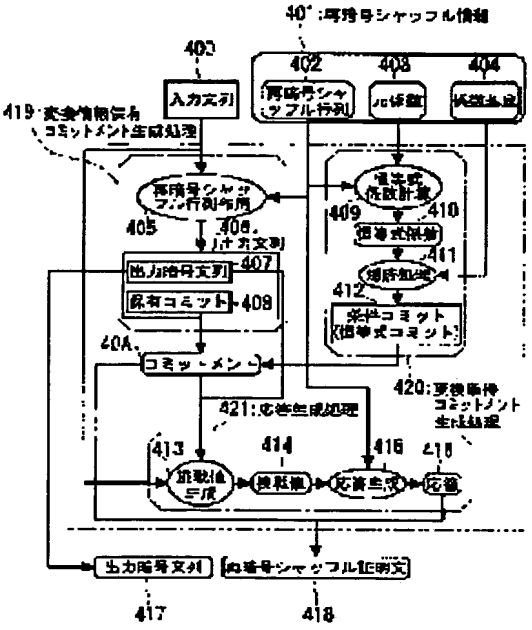
(51)Int.Cl. H04L 9/08
G09C 1/00
G09C 1/04

(21)Application number : 2000-059091
(22)Date of filing : 03.03.2000

(71)Applicant : NEC CORP
(72)Inventor : FURUKAWA JUN

(54) METHOD AND DEVICE FOR AUTHENTICATED RE-ENCIPHERMENT SHUFFLE, METHOD AND DEVICE FOR VERIFYING RE- ENCIPHERMENT SHUFFLE AND METHOD, AND DEVICE FOR GENERATING INPUT SENTENCE STRING AND RECORDING MEDIUM

(57)Abstract:
PROBLEM TO BE SOLVED: To provide certified re-encipherment shuffle having a certificating method and a verifying method in which calculation quantity can be made proportional to the number of input cipher texts and the calculation quantity can be made small, and to provided a method for verifying this certified re-encipherment shuffle.
SOLUTION: Re-encipherment is expressed as a kind of general conversion, and the authentication of re-encipherment shuffle can be constituted of the a



(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号
特開2001－251289
（P2001－251289A）
(43)公開日　平成13年９月14日(2001.9.14)

(51)Int.Cl.'		識別記号	FI		テーマコード'(参考)
H04L	9/08		G09C	1/00	620A　5J104
G09C	1/00	620			640Z　9A001
		640		1/04	
	1/04		H04L	9/00	601C

審査請求　有　請求項の数54　ＯＬ　（全 83 頁）

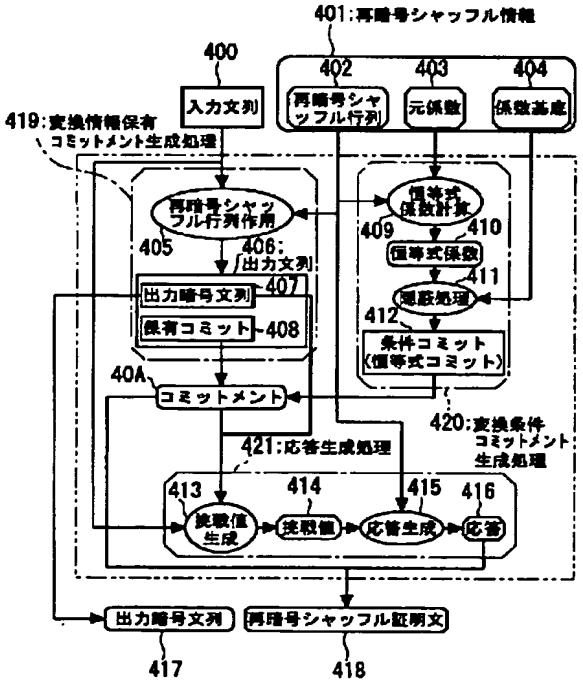
(21)出願番号　特願2000－59091(P2000－59091)
(22)出願日　平成12年３月３日(2000.3.3)

(71)出願人　000004237
日本電気株式会社
東京都港区芝五丁目７番１号
(72)発明者　古川　潤
東京都港区芝五丁目７番１号　日本電気株式会社内
(74)代理人　100080816
弁理士　加藤　朝道

Ｆターム(参考)　5J104 AA01 AA18 EA16 JA23 NA09
NA12 NA29 NA32
9A001 BB02 BB03 BB04 EE03 FF01
GG01 GG05 GG22 JJ18 KK56
LL03

(54)【発明の名称】　証明付再暗号シャッフル方法と装置、再暗号シャッフル検証方法と装置、入力文列生成方法と装置及び記録媒体

(57)【要約】
【課題】計算量が入力暗号文の数に比例しかつその計算量が少ない証明生成および検証方法をもつ証明付再暗号シャッフルおよびその検証方法の考案。
【解決手段】再暗号シャッフルをより一般的な変換の一種として表現し、この変換情報を保有していることの証明と変換の満たす条件の証明の二つを合わせて再暗号シャッフルの証明を構成するものであり、二種の証明は、それぞれ入力暗号文数に依存せず短く、変換情報保有の証明は、挑戦値から上記変換に依存して応答を生成するため、応答と挑戦値の関係に変換の満たす条件が反映され、挑戦値に依存しない応答と挑戦値の関係式が存在し、これが成り立つことから変換の満たす条件を証明する。証明すべき条件として、再暗号シャッフルに対応する変換の満たす条件を選べば両証明をもってして再暗号シャッフルの証明を構成できる。



1

【特許請求の範囲】

【請求項１】複数の暗号文と一つまたは複数の公開鍵とからなる入力文列と、再暗号シャッフル情報とを入力し、前記暗号文に対して順番の並び替えと前記公開鍵による再暗号化とを施した出力暗号文列と、上記処理に関する証明文である再暗号シャッフル証明文とを出力する証明付再暗号シャッフル方法において、

前記入力文列から出力暗号文列を生成するとともに、前記入力文列から前記出力暗号文への変換情報の保有に関するコミットメント（「変換情報保有コミットメント」という）を生成する変換情報保有コミットメント生成ステップと、

前記変換の満たす条件に関するコミットメント（「変換条件コミットメント」という）を生成する、変換条件コミットメント生成ステップと、

再暗号シャッフル情報と挑戦値とから応答を生成する、応答生成ステップと、を含み、

前記変換情報保有コミットメントと前記変換条件コミットメントと前記応答とを前記再暗号シャッフル証明文として出力し、

前記再暗号シャッフル情報は、入力暗号文の並び替え方と、再暗号化に用いた変数と、乱数とを含む、ことを特徴とする証明付再暗号シャッフル方法。

【請求項２】入力文列と、出力暗号文列と、再暗号シャッフル証明文が入力され、受理または不受理である検証結果を出力する再暗号シャッフル検証方法において、

前記入力文列と、前記出力暗号文列と、前記入力文列から前記出力暗号文への変換情報の保有に関する変換情報保有コミットメントと、応答と、挑戦値とから、前記入力文列から前記出力暗号文列への変換情報を保有していることを検証する、変換情報保有検証ステップと、

前記変換の満たす条件に関する変換条件コミットメントと、前記応答と、前記挑戦値とから、前記入力文列から前記出力暗号文列への変換の満たす条件を検証する、変換条件検証ステップと、

を含み、

前記変換情報保有検証ステップと前記変換条件検証ステップの検証がともに受理されたら、再暗号シャッフル検証結果として受理を、それ以外は不受理を出力する、ことを特徴とする再暗号シャッフル検証方法。

【請求項３】請求項１に記載の証明付再暗号シャッフル方法において、

前記変換情報保有コミットメント生成ステップは、前記出力暗号文列および前記変換情報保有コミットメントを、前記入力文列を基底として、再暗号化に用いた変数および並び替えに対応する値および乱数とを表現とした表現値として生成し、

前記変換条件コミットメント生成ステップは、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式の係数を、再

2

暗号シャッフル情報から積和演算のみを用いて生成し、

前記恒等式の係数、または前記係数の一部もしくは全てを、コミットしたものを前記変換条件コミットメントとし、

前記応答生成ステップは、前記応答を、再暗号シャッフル情報と挑戦値から積和演算のみで生成し、

前記表現は、表現値と基底を対応付けるものであって、無作為に与えられた表現値と基底から、これらを対応付ける表現を計算することは、計算量的に困難とされており、

前記挑戦値は、前記入力文列と前記出力暗号文列とコミットメント全てが決まった後に、無作為に決められる複数の成分、あるいは、前記入力文列と前記出力暗号文列と全てのコミットメントとを入力として挑戦値生成関数により出力される複数の成分であり、

前記挑戦値生成関数は、与えられた入力から複数の成分を出力するものであって、それらの出力からは入力を求めること、出力成分間の関係を意図して入力を決定することが、計算量的に困難である関数である、

ことを特徴とする証明付再暗号シャッフル方法。

【請求項４】請求項２に記載の再暗号シャッフル検証方法において、

前記変換情報保有検証ステップは、前記出力暗号文列および変換情報保有コミットメントを基底として挑戦値を表現とする表現値と、前記入力文列を基底として応答を表現とする表現値が等しいことを検証し、

前記変換条件検証ステップは、前記変換条件コミットメントにより、入力された応答と挑戦値に関して、前記入力文列から出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式が成立することを検証する、ことを特徴とする再暗号シャッフル検証方法。

【請求項５】請求項３に記載の証明付再暗号シャッフル方法において、

前記変換条件コミットメント生成ステップは、前記入力文列から前記出力暗号文列への変換の満たす条件を記述し、かつ、前記応答と準応答と挑戦値との多項式である恒等式の係数を、前記再暗号シャッフル情報から積和演算を用いて生成し、これらの恒等式の係数、または、これら係数の一部または全てをコミットしたものと、準元係数、または、これら係数の一部または全てをコミットしたものと、を変換条件コミットメントとして生成し、

前記準応答は、再暗号シャッフル検証における、変換情報保有検証処理には使用されないものであり、応答と挑戦値との多項式であり、前記多項式の係数が準元係数であり、

前記応答生成ステップは、前記挑戦値より再暗号シャッフル情報を用いて応答と準応答との二種の応答を生成し、

前記再暗号シャッフル証明文は、前記変換情報保有コミ

ットメントと前記変換条件コミットメントと前記応答と前記準応答とを含む、
ことを特徴とする証明付再暗号シャッフル方法。
【請求項６】請求項４に記載の再暗号シャッフル検証方法において、
前記変換条件検証ステップは、前記変換条件コミットメントにより、入力された応答と準応答と挑戦値に関して、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と準応答と挑戦値の多項式である恒等式が成立することを検証し、準元係数をコミットしたものと応答と挑戦値とより、前記準応答の正当性を検証する、ことを特徴とする再暗号シャッフル検証方法。
【請求項７】請求項１に記載の証明付再暗号シャッフル方法において、
前記変換情報保有コミットメント生成ステップは、前記出力暗号文列および前記変換情報保有コミットメントを、前記入力文列を基底として、再暗号化に用いた変数および並び替えに対応する値および乱数とを表現とした表現値として生成し、
前記変換条件コミットメント生成ステップは、複数個の変換条件コミットメント生成ステップよりなり、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式の係数を、再暗号シャッフル情報から積和演算を用いて生成し、前記恒等式の係数、前記係数の一部もしくは全てを、コミットしたものを前記変換条件コミットメントとする変換条件コミットメント生成ステップと、
前記入力文列から前記出力暗号文列への変換の満たす条件を記述し、かつ、前記応答と準応答と挑戦値との多項式である恒等式の係数を、前記再暗号シャッフル情報から積和演算を用いて生成し、これらの恒等式の係数または、これら係数の一部または全てをコミットしたものと、準元係数または、これら係数の一部または全てをコミットしたものと、を変換条件コミットメントとして生成する変換条件コミットメント生成ステップと、の両ステップの一方または両方を複数含み、
前記応答生成ステップは、前記応答と、前記変換条件コミットメント生成処理に応じて複数個の準応答を生成し、
前記再暗号シャッフル証明文は、複数の前記変換条件コミットメントと、このコミットメントに対応する準応答と、前記応答と、前記変換情報保有コミットメントとを含む、ことを特徴とする証明付再暗号シャッフル方法。
【請求項８】請求項２に記載の再暗号シャッフル検証方法において、
前記変換情報保有検証ステップは、前記出力暗号文列および変換情報保有コミットメントを基底として挑戦値を表現とする表現値と、前記入力文列を基底として応答を表現とする表現値が等しいことを検証し、
前記変換条件検証ステップは、複数個の変換条件検証ステップよりなり、前記変換条件コミットメントにより、入力された応答と挑戦値に関して、前記入力文列から出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式が成立することを検証する検証ステップと、
前記変換条件コミットメントにより、入力された応答と準応答と挑戦値に関して、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と準応答と挑戦値の多項式である恒等式が成立することを検証し、準元係数をコミットしたものと応答と挑戦値とより、前記準応答の正当性を検証する検証ステップと、の両ステップの一方または両方を複数含む、ことを特徴とする再暗号シャッフル検証方法。
【請求項９】請求項１に記載の証明付再暗号シャッフル方法において、
前記変換情報保有コミットメント生成ステップは、複数個の変換情報保有コミットメント生成ステップよりなり、前記各変換情報保有コミットメント生成ステップは、前記出力暗号文列および前記変換情報保有コミットメントを、前記入力文列を基底として、再暗号化に用いた変数および並び替えに対応する値および乱数とを表現とした表現値として生成し、２番目以降の変換情報保有コミットメント生成処理は、１番目の変換情報保有コミットメント生成ステップと共通する出力の生成を省略し、
前記変換条件コミットメント生成ステップは、複数個の変換条件コミットメント生成ステップよりなり、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式の係数を、再暗号シャッフル情報から積和演算を用いて生成し、前記恒等式の係数、前記係数の一部もしくは全てを、コミットしたものを前記変換条件コミットメントとするステップと、
前記入力文列から前記出力暗号文列への変換の満たす条件を記述し、かつ、前記応答と準応答と挑戦値との多項式である恒等式の係数を、前記再暗号シャッフル情報から積和演算を用いて生成し、これらの恒等式の係数または、これら係数の一部または全てをコミットしたものと、準元係数または、これら係数の一部または全てをコミットしたものと、を変換条件コミットメントとして生成するステップと、の両ステップの一方または両方を複数含み、
前記応答生成ステップは、前記変換情報保有コミットメント生成ステップに応じて複数個の応答を生成し、前記変換条件コミットメント生成ステップに応じて複数個の準応答を生成し、
前記再暗号シャッフル証明文は、前記複数個の応答と複数個の知識のコミットメントと複数個の変換条件コミットメントとそれに対応する準応答とよりなるものであ

る、ことを特徴とする証明付再暗号シャッフル方法。

【請求項１０】請求項２に記載の再暗号シャッフル検証方法において、

前記変換情報保有検証ステップは、複数個の変換情報保有検証ステップよりなり、前記各変換情報保有検証ステップは、前記出力暗号文列および変換情報保有コミットメントを基底として挑戦値を表現とする表現値と、前記入力文列を基底として応答を表現とする表現値が等しいことを検証し、

前記変換条件検証ステップは、複数個の変換条件検証ステップよりなり、前記変換条件コミットメントにより、入力された応答と挑戦値に関して、前記入力文列から出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式が成立することを検証するステップと、

前記変換条件コミットメントにより、入力された応答と準応答と挑戦値に関して、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と準応答と挑戦値の多項式である恒等式が成立することを検証し、準元係数をコミットしたものと応答と挑戦値とより、前記準応答の正当性を検証するステップと、の両ステップの一方または両方を複数含む、ことを特徴とする再暗号シャッフル検証方法。

【請求項１１】請求項３または請求項５に記載の証明付再暗号シャッフル方法において、

前記変換条件コミットメント生成ステップにおける恒等式が、応答の各成分は挑戦値の多項式よりなり、前記多項式の一部の、一部の項の各２乗の和と、挑戦値の成分の一部の各２乗の和とが、挑戦値によらずに等しくなる関係を内包している、ものであることを特徴とする証明付再暗号シャッフル方法。

【請求項１２】請求項３または請求項５に記載の証明付再暗号シャッフル方法において、

前記変換条件コミットメント生成ステップにおける恒等式が、応答の各成分は挑戦値の多項式であるが、これらの多項式の一部の、一部の項の各３乗の和と、挑戦値の成分の一部の各３乗の和とが、挑戦値によらずに等しくなる関係を内包しているものであることを特徴とする証明付再暗号シャッフル方法。

【請求項１３】請求項７または請求項９に記載の証明付再暗号シャッフル方法において、

前記変換条件コミットメント生成ステップにおける複数個の恒等式が、応答の各成分は挑戦値の多項式よりなり、前記多項式の一部の、一部の項の各２乗の和と、挑戦値の成分の一部の各２乗の和とが、挑戦値によらずに等しくなる関係を内包しているものと、応答の各成分は挑戦値の多項式であるが、これらの多項式の一部の、一部の項の各３乗の和と、挑戦値の成分の一部の各３乗の和とが、挑戦値によらずに等しくなる関係を内包しているものの二つを含む、ことを特徴とする証明付再暗号シャッフル方法。

【請求項１４】証明付再暗号シャッフル装置に入力する入力文列を、その一部を疑似乱数、または、公開鍵と入力暗号文列とに疑似乱数による変換を受けた数値として生成する、ことを特徴とする入力文列生成方法。

【請求項１５】請求項１４に記載の入力文列生成方法において、入力暗号文列と公開鍵と疑似乱数を合わせて入力文列とする、ことを特徴とする入力文列生成方法。

【請求項１６】請求項１４に記載の入力文列生成方法において、公開鍵列により入力暗号文列を変換し、公開鍵列の正当性証明文を出力する入力文列生成方法であって、前記公開鍵列が、秘密鍵を分散所持する証明者が協力して生成した、同じ秘密鍵に対応する複数の公開鍵で、その各公開鍵の特定の成分からなる集合は、入力文列を含む入力により生成された疑似乱数である、ことを特徴とする入力文列生成方法。

【請求項１７】請求項１４に記載の入力文列生成方法において、秘密鍵を分散所持する証明者が協力して生成した同じ秘密鍵に対応する複数の公開鍵で、その各公開鍵の特定の成分からなる集合は、入力文列を含む入力により生成された疑似乱数である公開鍵列を生成し、これを構成する各公開鍵で各入力平文を暗号化し、かつそれぞれの公開鍵で暗号化したことを証明し、この入力暗号文列と公開鍵を合わせて入力文列を生成する、ことを特徴とする入力文列生成方法。

【請求項１８】与えられた入力から一意的に決定される疑似乱数数列を成分に持ち、かつ同じ秘密鍵に対応する、公開鍵を成分とする公開鍵列と、同じ秘密鍵に対応することの証明文とを、秘密鍵を分散所持する証明者が協力して生成する、ことを特徴とする証明付公開鍵列生成方法。

【請求項１９】複数の入力暗号文と一又は複数の公開鍵とからなる入力文列と、入力暗号文の並び替え方と、再暗号化に用いた変数と、乱数とを含む再暗号シャッフル情報と、を入力し、前記暗号文に対して順番の並び替えと前記公開鍵による再暗号化とを施した出力暗号文列と、再暗号シャッフル証明文とを出力する証明付再暗号シャッフル装置であって、

前記入力文列から出力暗号文列を生成するとともに、前記入力文列から出力暗号文への変換情報の保有に関するコミットメント（「変換情報保有コミットメント」という）を生成する変換情報保有コミットメント生成部と、

前記変換の満たす条件に関するコミットメント（「変換条件コミットメント」という）を生成する、変換条件コミットメント生成部と、

再暗号シャッフル情報と挑戦値とから応答を生成する応答生成部と、

を備え、

前記変換情報保有コミットメントと前記変換条件コミットメントと前記応答とを前記再暗号シャッフル証明文と

して出力する、ことを特徴とする証明付再暗号シャッフル装置。

【請求項２０】複数の入力暗号文と一又は複数の公開鍵とからなる入力文列と、再暗号シャッフル情報とを入力し、前記暗号文に対して順番の並び替えと前記公開鍵による再暗号化とを施した出力暗号文列と、再暗号シャッフル証明文とを出力する証明付再暗号シャッフル装置に入力される前記入力文列と、

前記証明付再暗号シャッフル装置から出力される前記出力暗号文列と、

前記証明付再暗号シャッフル装置から出力される、前記入力文列から前記出力暗号文への変換情報の保有に関する変換情報保有コミットメントと、前記変換を満する条件に関する変換条件コミットメントと、応答とを含む再暗号シャッフル証明文と、

を入力とし、受理または不受理である検証結果を出力する再暗号シャッフル検証装置であって、

前記入力文列と、前記出力暗号文列と、前記変換情報保有コミットメントと、応答と、挑戦値とに基づき、前記入力文列から前記出力暗号文列への変換情報を保有していることを検証する、変換情報保有検証部と、

前記変換条件コミットメントと、前記応答と、前記挑戦値とに基づき、前記入力文列から前記出力暗号文列への変換の満たす条件を検証する、変換条件検証部と、

を備え、

前記変換情報保有検証部と前記変換条件検証部における検証がともに受理された場合に、再暗号シャッフル検証結果として受理を出力し、それ以外は不受理を出力する、ことを特徴とする再暗号シャッフル検証装置。

【請求項２１】請求項１９に記載の証明付再暗号シャッフル装置において、

前記変換情報保有コミットメント生成部が、前記出力暗号文列および前記変換情報保有コミットメントを、前記入力文列を基底として、再暗号化に用いた変数、および並び替えに対応する値、および、乱数とを表現とした表現値として生成する手段を備え、

前記変換条件コミットメント生成部が、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式の係数を、前記再暗号シャッフル情報から、積和演算を用いて生成し、前記恒等式の係数、前記係数の一部もしくは全てをコミットしたものを、前記変換条件コミットメントとして出力する手段を備え、

前記応答生成部が、前記応答を、前記再暗号シャッフル情報と、前記入力文列と前記出力暗号文列とコミットメント全てが決まった後に、無作為に決められる複数の成分、あるいは、前記入力文列と前記出力暗号文列と全てのコミットメントとを入力として挑戦値生成関数により出力される複数の成分である挑戦値から積和演算を用いて生成する手段を備えたことを特徴とする証明付再暗号シャッフル装置。

【請求項２２】請求項２０に記載の再暗号シャッフル検証装置において、

前記変換情報保有検証部が、前記出力暗号文列および変換情報保有コミットメントを基底として挑戦値を表現とする表現値と、前記入力文列を基底として応答を表現とする表現値が等しいことを検証する手段を備え、

前記変換条件検証部が、前記変換条件コミットメントにより、入力された応答と挑戦値に関して、前記入力文列から出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式が成立することを検証する手段を備えた、ことを特徴とする再暗号シャッフル検証装置。

【請求項２３】請求項２１に記載の証明付再暗号シャッフル装置において、

前記変換条件コミットメント生成部が、前記入力文列から前記出力暗号文列への変換の満たす条件を記述し、かつ、前記応答と準応答と挑戦値との多項式である恒等式の係数を、前記再暗号シャッフル情報から積和演算を用いて生成する手段と、

前記恒等式の係数、または、これら係数の一部または全てをコミットしたものと、準元係数、または、これら係数の一部または全てをコミットしたものと、を変換条件コミットメントとして生成する手段と、を備え、

前記準応答は、これは応答と挑戦値との多項式であり、前記多項式の係数が準元係数であり、

前記応答生成部は、前記挑戦値より再暗号シャッフル情報を用いて応答と準応答との二種の応答を生成する手段を備え、

前記再暗号シャッフル証明文が、前記変換情報保有コミットメントと前記変換条件コミットメントと前記応答と前記準応答よりなる、

ことを特徴とする証明付再暗号シャッフル装置。

【請求項２４】請求項２２に記載の再暗号シャッフル検証装置において、

前記変換条件検証部が、前記変換条件コミットメントにより、入力された応答と準応答と挑戦値に関して、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と準応答と挑戦値の多項式である恒等式が成立することを検証し、準元係数をコミットしたものと応答と挑戦値とより、前記準応答の正当性を検証する、ことを特徴とする再暗号シャッフル検証装置。

【請求項２５】請求項１９に記載の証明付再暗号シャッフル装置において、

前記変換情報保有コミットメント生成部が、前記出力暗号文列および前記変換情報保有コミットメントを、前記入力文列を基底として、再暗号化に用いた変数および並び替えに対応する値および乱数とを表現とした表現値として生成する手段を備え、

前記変換条件コミットメント生成部を複数備え、前記入

力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式の係数を、再暗号シャッフル情報から積和演算を用いて生成し、前記恒等式の係数、前記係数の一部もしくは全てを、コミットしたものを前記変換条件コミットメントとする前記変換条件コミットメント生成部と、
前記入力文列から前記出力暗号文列への変換の満たす条件を記述し、かつ、前記応答と準応答と挑戦値との多項式である恒等式の係数を、前記再暗号シャッフル情報から積和演算を用いて生成し、これらの恒等式の係数または、これら係数の一部または全てをコミットしたものと、準元係数または、これら係数の一部または全てをコミットしたものと、を変換条件コミットメントとして生成する変換条件コミットメント生成部と、の両生成部の一方または両方を含み、
前記応答生成部が、前記応答と、前記複数の変換条件コミットメント生成部に応じて複数個の準応答を生成する手段を備え、
前記再暗号シャッフル証明文が、複数の変換条件コミットメントとそれに対応する準応答と、応答と変換情報保有コミットメントよりなる、ことを特徴とする証明付再暗号シャッフル装置。
【請求項２６】請求項２０に記載の再暗号シャッフル検証装置において、
前記変換情報保有検証部が、前記出力暗号文列および変換情報保有コミットメントを基底として挑戦値を表現とする表現値と、前記入力文列を基底として応答を表現とする表現値が等しいことを検証し、
前記変換条件検証部が、複数個の変換条件検証部よりなり、
前記変換条件コミットメントにより、入力された応答と挑戦値に関して、前記入力文列から出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式が成立することを検証する検証部と、
前記変換条件コミットメントにより、入力された応答と準応答と挑戦値に関して、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と準応答と挑戦値の多項式である恒等式が成立することを検証し、準元係数をコミットしたものと応答と挑戦値とより、前記準応答の正当性を検証する検証部と、の両検証部の一方または両方を複数含むことを特徴とする再暗号シャッフル検証装置。
【請求項２７】請求項１９に記載の証明付再暗号シャッフル装置において、
前記変換情報保有コミットメント生成部が、複数の変換情報保有コミットメント生成部よりなり、前記出力暗号文列および前記変換情報保有コミットメントを、前記入力文列を基底として、再暗号化に用いた変数および並び替えに対応する値および乱数とを表現とした表現値として生成し、２番目以降の変換情報保有コミットメント生成処理は１番目の変換情報保有コミットメント生成処理と共通する出力の生成を省略し、
前記変換条件コミットメント生成部が、複数の変換条件コミットメント生成部よりなり、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式の係数を、再暗号シャッフル情報から積和演算を用いて生成し、前記恒等式の係数、前記係数の一部もしくは全てを、コミットしたものを前記変換条件コミットメントとする変換条件コミットメント生成部と、
前記入力文列から前記出力暗号文列への変換の満たす条件を記述し、かつ、前記応答と準応答と挑戦値との多項式である恒等式の係数を、前記再暗号シャッフル情報から積和演算を用いて生成し、これらの恒等式の係数または、これら係数の一部または全てをコミットしたものと、準元係数または、これら係数の一部または全てをコミットしたものと、を変換条件コミットメントとして生成する変換条件コミットメント生成部と、の両生成部の一方または両方を複数含み、
前記応答生成処理が、前記複数の変換情報保有コミットメント生成部の出力に応じて複数個の応答を生成し、前記複数の変換条件コミットメント生成部の出力に応じて複数個の準応答を生成する手段を備え、
前記再暗号シャッフル証明文は、前記複数個の応答と複数個の知識のコミットメントと複数個の変換条件コミットメントとそれに対応する準応答よりなる、ことを特徴とする証明付再暗号シャッフル装置。
【請求項２８】請求項２０に記載の再暗号シャッフル検証装置において、
前記変換情報保有検証部が、複数の変換情報保有検証部よりなり、前記出力暗号文列および変換情報保有コミットメントを基底として挑戦値を表現とする表現値と、前記入力文列を基底として応答を表現とする表現値が等しいことを検証し、
前記変換条件検証部が、複数の変換条件検証部よりなり、前記変換条件コミットメントにより、入力された応答と挑戦値に関して、前記入力文列から出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式が成立することを検証する変換条件検証部と、
前記変換条件コミットメントにより、入力された応答と準応答と挑戦値に関して、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と準応答と挑戦値の多項式である恒等式が成立することを検証し、準元係数をコミットしたものと応答と挑戦値とより、前記準応答の正当性を検証する変換条件検証部と、の両検証部の一方または両方を複数含むことを特徴とする再暗号シャッフル検証装置。
【請求項２９】請求項２１または請求項２３に記載の証明付再暗号シャッフル装置において、

前記変換条件コミットメント生成部における恒等式が、応答の各成分は、挑戦値の多項式よりなり、前記多項式の一部の、一部の項の各２乗の和と、挑戦値の成分の一部の各２乗の和とが、挑戦値によらずに等しくなる関係を内包している、ものであることを特徴とする証明付再暗号シャッフル装置。

【請求項３０】請求項２１または請求項２３に記載の証明付再暗号シャッフル装置において、
前記変換条件コミットメント生成部における恒等式が、応答の各成分は挑戦値の多項式であるが、これらの多項式の一部の、一部の項の各３乗の和と、挑戦値の成分の一部の各３乗の和とが、挑戦値によらずに等しくなる関係を内包しているものである、ことを特徴とする証明付再暗号シャッフル装置。

【請求項３１】請求項２５または請求項２７に記載の証明付再暗号シャッフル装置において、
前記変換条件コミットメント生成部における複数個の恒等式が、応答の各成分は挑戦値の多項式よりなり、前記多項式の一部の、一部の項の各２乗の和と、挑戦値の成分の一部の各２乗の和とが、挑戦値によらずに等しくなる関係を内包しているものと、応答の各成分は挑戦値の多項式であるが、これらの多項式の一部の、一部の項の各３乗の和と、挑戦値の成分の一部の各３乗の和とが、挑戦値によらずに等しくなる関係を内包しているものを含む、ことを特徴とする再暗号シャッフル装置。

【請求項３２】請求項１９記載の前記証明付再暗号シャッフル装置に入力する前記入力文列を、その一部を、疑似乱数または、公開鍵と入力暗号文列とに疑似乱数による変換を受けた数値として生成する入力文列生成装置。

【請求項３３】請求項３２に記載の入力文列生成装置において、入力暗号文列と公開鍵と疑似乱数を合わせて入力文列とする入力文列生成装置。

【請求項３４】請求項３２に記載の入力文列生成装置において、公開鍵列により入力暗号文列を変換し、公開鍵列の正当性証明文を出力する入力文列生成装置であって、前記公開鍵列が、秘密鍵を分散所持する証明者が協力して生成した、同じ秘密鍵に対応する複数の公開鍵で、その各公開鍵の特定の成分からなる集合は、入力文列を含む入力により生成された疑似乱数である、ことを特徴とする入力文列生成装置。

【請求項３５】請求項３２に記載の入力文列生成装置において、秘密鍵を分散所持する証明者が協力して生成した同じ秘密鍵に対応する複数の公開鍵で、その各公開鍵の特定の成分からなる集合は、入力文列を含む入力により生成された疑似乱数である公開鍵列を生成し、これを構成する各公開鍵で各入力平文を暗号化し、かつそれぞれの公開鍵で暗号化したことを証明し、この入力暗号文列と公開鍵を合わせて入力文列を生成する入力文列生成装置。

【請求項３６】与えらた入力から一意的に決定される疑似乱数数列を成分に持ち、かつ同じ秘密鍵に対応する、公開鍵を成分とする公開鍵列と、同じ秘密鍵に対応することの証明文とを、秘密鍵を分散所持する証明者が協力して生成する手段を備えたことを特徴とする証明付公開鍵列生成装置。

【請求項３７】複数の入力暗号文と一又は複数の公開鍵とからなる入力文列と、入力暗号文の並び替え方と、再暗号化に用いた変数と、乱数とを含む再暗号シャッフル情報と、を入力し、前記暗号文に対して、順番の並び替えと、前記公開鍵による再暗号化を施した出力暗号文列と、再暗号シャッフル証明文とを出力する再暗号シャッフル装置であって、
（ａ）前記入力文列から出力暗号文列を生成するとともに、前記入力文列から出力暗号文への変換情報の保有に関するコミットメント（「変換情報保有コミットメント」という）を生成する変換情報保有コミットメント生成処理と、
（ｂ）前記変換の満たす条件に関するコミットメント（「変換条件コミットメント」という）を生成する、変換条件コミットメント生成処理と、
（ｃ）再暗号シャッフル情報と挑戦値とから応答を生成する応答生成処理と、
（ｄ）前記変換情報保有コミットメントと前記変換条件コミットメントと前記応答とを前記再暗号シャッフル証明文として出力する処理、
の前記（ａ）乃至（ｄ）の処理を証明付再暗号シャッフル装置を構成するコンピュータで実行させるためのプログラムを記録した記録媒体。

【請求項３８】入力文列と、証明付再暗号シャッフル装置から出力される出力暗号文列と、再暗号シャッフル装置から出力される、前記入力文列から前記出力暗号文への変換情報の保有に関する変換情報保有コミットメントと、前記変換を満する条件に関する変換条件コミットメントと、応答とを含む再暗号シャッフル証明文と、を入力とし、受理または不受理である検証結果を出力する再暗号シャッフル検証装置であって、
（ａ）前記入力文列と、前記出力暗号文列と、前記変換情報保有コミットメントと、応答と、挑戦値とより、前記入力文列から前記出力暗号文列への変換情報を保有していることを検証する、変換情報保有検証処理と、
（ｂ）前記変換条件コミットメントと、前記応答と、前記挑戦値とから、前記入力文列から前記出力暗号文列への変換の満たす条件を検証する、変換条件検証処理と、
（ｃ）前記変換情報保有検証処理と前記変換条件検証処理がともに受理された場合に、再暗号シャッフル検証結果として受理を出力し、それ以外は不受理を出力する処理、
の前記（ａ）乃至（ｃ）の処理を再暗号シャッフル検証装置を構成するコンピュータで実行させるためのプログラムを記録した記録媒体。

【請求項３９】請求項３７に記載の記憶媒体において、前記変換情報保有コミットメント生成処理が、
前記出力暗号文列および前記変換情報保有コミットメントを、前記入力文列を基底として、再暗号化に用いた変数、および並び替えに対応する値、および、乱数とを表現とした表現値として生成し、
前記変換条件コミットメント生成処理が、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式の係数を、前記再暗号シャッフル情報から、積和演算を用いて生成し、前記恒等式の係数、前記係数の一部もしくは全てをコミットしたものを、前記変換条件コミットメントとして出力し、
前記応答生成処理は、前記応答を、前記再暗号シャッフル情報と、前記入力文列と前記出力暗号文列とコミットメント全てが決まった後に、無作為に決められる複数の成分、あるいは、前記入力文列と前記出力暗号文列と全てのコミットメントとを入力として挑戦値生成関数により出力される複数の成分である挑戦値から積和演算を用いて生成する、
前記各処理を前記証明付再暗号シャッフル装置を構成するコンピュータで実行させるためのプログラムを記録した記録媒体。

【請求項４０】請求項３８に記載の記憶媒体において、前記変換情報保有検証処理は、前記出力暗号文列および変換情報保有コミットメントを基底として挑戦値を表現とする表現値と、前記入力文列を基底として応答を表現とする表現値が等しいことを検証し、
前記変換条件検証処理は、前記変換条件コミットメントにより、入力された応答と挑戦値に関して、前記入力文列から出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式が成立することを検証する、
前記各処理を前記再暗号シャッフル検証装置を構成するコンピュータで実行させるためのプログラムを記録した記録媒体。

【請求項４１】請求項３７に記載の記憶媒体において、前記変換条件コミットメント生成処理が、前記入力文列から前記出力暗号文列への変換の満たす条件を記述し、かつ、前記応答と準応答と挑戦値との多項式である恒等式の係数を、前記再暗号シャッフル情報から積和演算を用いて生成し、
前記恒等式の係数、または、これら係数の一部または全てをコミットしたものと、準元係数、または、これら係数の一部または全てをコミットしたものと、を変換条件コミットメントとして生成し、
前記準応答は、これは応答と挑戦値との多項式であり、前記多項式の係数が準元係数であり、
前記応答生成処理は、前記挑戦値より再暗号シャッフル情報を用いて応答と準応答との二種の応答を生成し、
前記再暗号シャッフル証明文として、前記変換情報保有コミットメントと前記変換条件コミットメントと前記応答と前記準応答を出力する、
前記各処理を前記証明付再暗号シャッフル装置を構成するコンピュータで実行させるためのプログラムを記録した記録媒体。

【請求項４２】請求項４０に記載の記録媒体において、前記変換条件検証処理が、前記変換条件コミットメントにより、入力された応答と準応答と挑戦値に関して、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と準応答と挑戦値の多項式である恒等式が成立することを検証し、準元係数をコミットしたものと応答と挑戦値とより、前記準応答の正当性を検証する、
前記変換条件検証処理を、前記再暗号シャッフル検証装置を構成するコンピュータで実行させるためのプログラムを記録した記録媒体。

【請求項４３】請求項３７に記載の記録媒体において、前記変換情報保有コミットメント生成処理が、前記出力暗号文列および前記変換情報保有コミットメントを、前記入力文列を基底として、再暗号化に用いた変数および並び替えに対応する値および乱数とを表現とした表現値として生成し、
複数の前記変換条件コミットメント生成処理を備え、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式の係数を、再暗号シャッフル情報から積和演算を用いて生成し、前記恒等式の係数、前記係数の一部もしくは全てを、コミットしたものを前記変換条件コミットメントとする前記変換条件コミットメント生成処理と、
前記入力文列から前記出力暗号文列への変換の満たす条件を記述し、かつ、前記応答と準応答と挑戦値との多項式である恒等式の係数を、前記再暗号シャッフル情報から積和演算を用いて生成し、これらの恒等式の係数または、これら係数の一部または全てをコミットしたものと、準元係数または、これら係数の一部または全てをコミットしたものと、を変換条件コミットメントとして生成する変換条件コミットメント生成処理と、をからなり、
前記応答生成処理は、前記応答と、前記複数の変換条件コミットメント生成部に応じて複数個の準応答を生成し、
前記再暗号シャッフル証明文として複数の変換条件コミットメントとそれに対応する準応答と、応答と変換情報保有コミットメントを出力する、
前記各処理を、前記証明付再暗号シャッフル装置を構成するコンピュータで実行させるためのプログラムを記録した記録媒体。

【請求項４４】請求項３８に記載の記録媒体において、前記変換情報保有検証処理が、前記出力暗号文列および

変換情報保有コミットメントを基底として挑戦値を表現とする表現値と、前記入力文列を基底として応答を表現とする表現値が等しいことを検証し、
前記変換条件検証処理が、複数の変換条件検証処理よりなり、
前記変換条件コミットメントにより、入力された応答と挑戦値に関して、前記入力文列から出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式が成立することを検証する検証処理と、
前記変換条件コミットメントにより、入力された応答と準応答と挑戦値に関して、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と準応答と挑戦値の多項式である恒等式が成立することを検証し、準元係数をコミットしたものと応答と挑戦値とより、前記準応答の正当性を検証する検証処理と、からなり、
前記各処理を、前記再暗号シャッフル検証装置を構成するコンピュータで実行させるためのプログラムを記録した記録媒体。

【請求項４５】請求項３７に記載の記録媒体において、
前記変換情報保有コミットメント生成処理が、複数の変換情報保有コミットメント生成処理よりなり、前記出力暗号文列および前記変換情報保有コミットメントを、前記入力文列を基底として、再暗号化に用いた変数および並び替えに対応する値および乱数とを表現とした表現値として生成し、２番目以降の変換情報保有コミットメント生成処理は１番目の変換情報保有コミットメント生成処理と共通する出力の生成を省略し、
前記変換条件コミットメント生成処理が、複数の変換条件コミットメント生成処理よりなり、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式の係数を、再暗号シャッフル情報から積和演算を用いて生成し、前記恒等式の係数、前記係数の一部もしくは全てを、コミットしたものを前記変換条件コミットメントとする変換条件コミットメント生成処理と、前記入力文列から前記出力暗号文列への変換の満たす条件を記述し、かつ、前記応答と準応答と挑戦値との多項式である恒等式の係数を、前記再暗号シャッフル情報から積和演算を用いて生成し、これらの恒等式の係数または、これら係数の一部または全てをコミットしたものと、準元係数または、これら係数の一部または全てをコミットしたものと、を変換条件コミットメントとして生成する変換条件コミットメント生成処理と、を含み、
前記応答生成処理は、前記複数の変換情報保有コミットメント生成部の出力に応じて複数個の応答を生成し、前記複数の変換条件コミットメント生成部の出力に応じて複数個の準応答を生成し、
前記再暗号シャッフル証明文として前記複数個の応答と複数個の知識のコミットメントと複数個の変換条件コミットメントとそれに対応する準応答とを出力する、
前記各処理を、前記証明付再暗号シャッフル装置を構成するコンピュータで実行させるためのプログラムを記録した記録媒体。

【請求項４６】請求項３８に記載の記録媒体において、
前記変換情報保有検証処理が、複数の変換情報保有検証処理よりなり、前記出力暗号文列および変換情報保有コミットメントを基底として挑戦値を表現とする表現値と、前記入力文列を基底として応答を表現とする表現値が等しいことを検証し、
前記変換条件検証処理が、複数の変換条件検証処理よりなり、前記変換条件コミットメントにより、入力された応答と挑戦値に関して、前記入力文列から出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式が成立することを検証する変換条件検証処理と、
前記変換条件コミットメントにより、入力された応答と準応答と挑戦値に関して、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と準応答と挑戦値の多項式である恒等式が成立することを検証し、準元係数をコミットしたものと応答と挑戦値とより、前記準応答の正当性を検証する変換条件検証処理と、を含み、
前記各処理を、前記再暗号シャッフル検証装置を構成するコンピュータで実行させるためのプログラムを記録した記録媒体。

【請求項４７】請求項３９または請求項４１に記載の記録媒体において、
前記変換条件コミットメント生成処理における恒等式が、応答の各成分は、挑戦値の多項式よりなり、前記多項式の一部の、一部の項の各２乗の和と、挑戦値の成分の一部の各２乗の和とが、挑戦値によらずに等しくなる関係を内包しており、
前記変換条件コミットメント生成処理を、前記証明付再暗号シャッフル装置を構成するコンピュータで実行させるためのプログラムを記録した記録媒体。

【請求項４８】請求項３９または請求項４１に記載の記録媒体において、
前記変換条件コミットメント生成処理における恒等式が、応答の各成分は挑戦値の多項式であるが、これらの多項式の一部の、一部の項の各３乗の和と、挑戦値の成分の一部の各３乗の和とが、挑戦値によらずに等しくなる関係を内包しており、
前記変換条件コミットメント生成処理を、前記証明付再暗号シャッフル装置を構成するコンピュータで実行させるためのプログラムを記録した記録媒体。

【請求項４９】請求項４３または請求項４５に記載の記録媒体において、
前記変換条件コミットメント生成処理における複数個の恒等式が、応答の各成分は挑戦値の多項式よりなり、前

記多項式の一部の、一部の項の各２乗の和と、挑戦値の成分の一部の各２乗の和とが、挑戦値によらずに等しくなる関係を内包しているものと、応答の各成分は挑戦値の多項式であるが、これらの多項式の一部の、一部の項の各３乗の和と、挑戦値の成分の一部の各３乗の和とが、挑戦値によらずに等しくなる関係を内包しているものの二つを含み、

前記変換条件コミットメント生成処理を、前記証明付再暗号シャッフル装置を構成するコンピュータで実行させるためのプログラムを記録した記録媒体。

【請求項５０】請求項３７に記載の記録媒体において、前記証明付再暗号シャッフル装置に入力する前記入力文列を、その一部を、疑似乱数または、公開鍵と入力暗号文列とに疑似乱数による変換を受けた数値として生成する入力文列生成処理を、コンピュータで実行させるためのプログラムを記録した記録媒体。

【請求項５１】請求項５０に記載の記録媒体において、入力暗号文列と公開鍵と疑似乱数を合わせて入力文列とする入力文列生成処理をコンピュータで実行させるためのプログラムを記録した記録媒体。

【請求項５２】請求項５０に記載の記録媒体において、公開鍵列により入力暗号文列を変換し、公開鍵列の正当性証明文を出力する入力文列生成処理であって、前記公開鍵列が、秘密鍵を分散所持する証明者が協力して生成した、同じ秘密鍵に対応する複数の公開鍵であり、前記各公開鍵の特定の成分からなる集合は、入力文列を含む入力により生成された疑似乱数とする、入力文列生成処理をコンピュータで実行させるためのプログラムを記録した記録媒体。

【請求項５３】請求項５０に記載の記録媒体において、秘密鍵を分散所持する証明者が協力して生成した同じ秘密鍵に対応する複数の公開鍵で、その各公開鍵の特定の成分からなる集合は、入力文列を含む入力により生成された疑似乱数である公開鍵列を生成し、これを構成する各公開鍵で各入力平文を暗号化し、かつそれぞれの公開鍵で暗号化したことを証明し、この入力暗号文列と公開鍵を合わせて入力文列を生成する入力文列生成処理をコンピュータで実行させるためのプログラムを記録した記録媒体。

【請求項５４】与えらた入力から一意的に決定される疑似乱数数列を成分に持ち、かつ同じ秘密鍵に対応する、公開鍵を成分とする公開鍵列と、同じ秘密鍵に対応することの証明文とを、秘密鍵を分散所持する証明者が協力して生成する証明付公開鍵列生成処理を、コンピュータで実行させるためのプログラムを記録した記録媒体。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、匿名通信路の構成などに使われる、入出力暗号文の一対一の対応関係を秘匿しつつ一対一の対応の存在を保証する再暗号シャッフル技術および再暗号シャッフル検証技術に関する。

【０００２】

【従来の技術】［従来の技術(1)］従来の証明付再暗号シャッフルの技術について、例えば特開平08-263575号公報（「文献１」という）の記載が参照される。図１に、該文献１に記載される構成を示す。なお、本願添付図面中で、合流する矢印は、矢印の元の情報が、全て集まって矢印の先へ送られることを意味し、分岐する矢印は矢印の元の情報全てまたは一部が、それぞれの矢印の先へ送られることを意味する。また破線は使用する入力文列生成法に依存することを示す。

【０００３】図１において、１６０個ある疑似出力暗号文列は零知識証明のコミットメントである。挑戦値は、入出力暗号文とコミットメントより生成して、応答は挑戦値のビット値に応じて実線または点線の矢印で示される入力暗号文列または出力暗号文列から疑似出力暗号文列への写像の明示である。

【０００４】図１に示すように、複数のElgamal入力暗号100の順序を並び替えて再び暗号化して出力する手法が紹介されている。このような処理を、「暗号シャッフル」という。この処理が正当であることを保証する為に同文献では以下の手法が紹介されている。並べ替えと再暗号の秘密乱数を毎回異なるものにして、再暗号シャッフルと同様の操作を安全変数(約160)の回数繰り返して疑似出力暗号文列を出力し、これを正当性証明のコミットメントとする。そしてこれら入出力暗号文とコミットメントのハッシュ値を挑戦値105として生成する。

【０００５】この挑戦値のビット列を上から順に読み、ビットが“1”の時は入力暗号文列から、“0”の時は出力暗号文列からの並び替え(並び替えを表す写像)と再暗号化(再び暗号化した時に使ったの乱数)の明示を応答106とする。

【０００６】以上のコミットメント、挑戦値、応答を再暗号シャッフルの証明文として出力する。以上においてハッシュ値のビット値に応じて対応関係を明示する方法をCut&Choose（カット・アンド・チューズ）法と言う。

【０００７】［従来の技術(２)］他の従来技術としては、阿部が、1999年電子情報通信学会情報セキュリティ技術報告書で発表した、「AMix-network on Permutation Networks」（「文献２」という）が参照される。この文献２では、例えば図２に示すように、一対の入力暗号文の置換200を繰り返して、全体として複数の入力暗号文の並び替えを実現する。各入力暗号文の置換の証明を、Cut&Choose法でない方法で構成することにより、ある数より小さい入力暗号文数の証明付再暗号シャッフルとしては効率の向上を達成している。すなわち、個々の入力暗号文を置換することによって入力暗号文列全体の並び替えを実現している。個々の置換の証明は効率の良いものであるが、置換を多くそろえる必要がある。

【０００８】

【発明が解決しようとする課題】しかしながら、上記した従来の技術は下記記載の問題点を有している。

【０００９】従来の技術(1)においては、コミットメントの生成のために安全変数(約160)の回数暗号シャッフルしなければならない。一回の再暗号シャッフルは、入力暗号文の数の2倍の冪乗剰余演算を行わねばならず、計算量が多い。

【００１０】また検証は、コミットメントを生成するのと同数の冪乗剰余演算を行わねばならず、計算量が多い。

【００１１】次に従来の技術(2)においては、一対の入力暗号文の置換とその証明のコミットメントは、合わせて、８回の冪乗剰余演算が必要である。

【００１２】この一置換当りの計算量は、従来の技術(1)の２入力暗号文あたりの計算量(=320)と比較すると小さいものの、入力暗号文全体のどのような並び替えでも実現できる回数にわたる一対の入力暗号文の置換が必要とされており、この数は、入力暗号文の数をnとすると、n logn-n+1である。

【００１３】このため、入力暗号文の数が増大すると、計算量が多くなる。

【００１４】したがって、本発明は、上記問題点に鑑みてなされたものであって、その目的は、入力暗号文数に依存せずに証明の計算量の短縮を図る方法及びシステム並びに記録媒体を提供することにある。

【００１５】本発明の他の目的は、検証処理を、証明同様に計算量の短縮を図る方法及びシステム並びに記録媒体を提供することにある。これ以外の本発明の目的、特徴、利点等は以下の実施の形態の記載から、当業者には直ちに明らかとされるであろう。

【００１６】

【課題を解決するための手段】前記目的を達成する本発明による証明付再暗号シャッフル方法は、複数の暗号文と一つまたは複数の公開鍵とからなる入力文列と、再暗号シャッフル情報とを入力し、前記暗号文に対して、順番の並び替えと、前記公開鍵による再暗号化を施した出力暗号文列と、上記処理に関する証明文である再暗号シャッフル証明文とを出力する証明付再暗号シャッフル方法において、前記入力文列から出力暗号文列を生成するとともに、前記入力文列から出力暗号文への変換情報の保有に関するコミットメント（「変換情報保有コミットメント」という）を生成する変換情報保有コミットメント生成ステップと、前記変換の満たす条件に関するコミットメント（「変換条件コミットメント」という）を生成する、変換条件コミットメント生成ステップと、再暗号シャッフル情報と挑戦値とから応答を生成する、応答生成ステップと、を含み、前記変換情報保有コミットメントと前記変換条件コミットメントと前記応答とを前記再暗号シャッフル証明文として出力し、前記再暗号シャッフル情報は、入力暗号文の並び替え方と、再暗号化に用いた変数と、乱数とからなる、ことを特徴とする。

【００１７】また本発明に係る再暗号シャッフル検証方法は、入力文列と、出力暗号文列と、再暗号シャッフル証明文が入力され、受理または不受理である検証結果を出力する再暗号シャッフル検証方法において、前記入力文列と、前記出力暗号文列と、変換情報保有コミットメントと、応答と、挑戦値とより、前記入力文列から前記出力暗号文列への変換情報を保有していることを検証する、変換情報保有検証ステップと、変換条件コミットメントと、前記応答と、前記挑戦値とから、前記入力文列から前記出力暗号文列への変換の満たす条件を検証する、変換条件検証ステップと、を含み、前記変換情報保有検証処理と前記変換条件検証処理の検証がともに受理されたら、再暗号シャッフル検証結果として受理を、それ以外は不受理を出力する、ことを特徴とする。

【００１８】本発明の証明付再暗号シャッフル装置は、複数の入力暗号文と一又は複数の公開鍵とからなる入力文列と、入力暗号文の並び替え方と、再暗号化に用いた変数と、乱数とを含む再暗号シャッフル情報と、を入力し、前記暗号文の順番を並び替え、前記公開鍵による再暗号化を施した出力暗号文列と、再暗号シャッフル証明文とを出力する証明付再暗号シャッフル装置であって、前記入力文列から出力暗号文列を生成するとともに、前記入力文列から出力暗号文への変換情報の保有に関するコミットメント（「変換情報保有コミットメント」という）を生成する変換情報保有コミットメント生成部と、前記変換の満たす条件に関するコミットメント（「変換条件コミットメント」という）を生成する、変換条件コミットメント生成部と、再暗号シャッフル情報と挑戦値とから応答を生成する応答生成部と、を備え、前記変換情報保有コミットメントと前記変換条件コミットメントと前記応答とを前記再暗号シャッフル証明文として出力する。

【００１９】本発明の再暗号シャッフル検証装置は、複数の入力暗号文と一又は複数の公開鍵とからなる入力文列と、入力暗号文の並び替え方と、再暗号化に用いた変数と、乱数とを含む再暗号シャッフル情報と、を入力し、前記暗号文の順番を並び替え、前記公開鍵による再暗号化を施した出力暗号文列と、再暗号シャッフル証明文とを出力する証明付再暗号シャッフル装置に入力される前記入力文列と、前記再暗号シャッフル装置から出力される前記出力暗号文列と、前記再暗号シャッフル装置から出力される前記再暗号シャッフル証明文とを入力とし、受理または不受理である検証結果を出力する再暗号シャッフル検証装置であって、前記入力文列と、前記出力暗号文列と、変換情報保有コミットメントと、応答と、挑戦値とより、前記入力文列から前記出力暗号文列への変換情報を保有していることを検証する、変換情報保有検証部と、前記変換条件コミットメントと、前記応答と、前記挑戦値とから、前記入力文列から前記出力暗

号文列への変換の満たす条件を検証する、変換条件検証部と、を備え、前記変換情報保有検証部と前記変換条件検証部における検証がともに受理された場合に、再暗号シャッフル検証結果として受理を出力し、それ以外は不受理を出力する。

【００２０】また本発明に係る入力文列生成方法は、証明付再暗号シャッフル装置に入力する入力文列を、その一部を疑似乱数または、公開鍵と入力暗号文列とに疑似乱数による変換を受けた数値として生成する。本発明においては、入力暗号文列と公開鍵と疑似乱数を合わせて入力文列としてもよい。

【００２１】本発明に係る記録媒体は、複数の入力暗号文と一又は複数の公開鍵とからなる入力文列と、入力暗号文の並び替え方と、再暗号化に用いた変数と、乱数とを含む再暗号シャッフル情報と、を入力し、前記暗号文に対して、順番の並び替えと、前記公開鍵による再暗号化を施した出力暗号文列と、再暗号シャッフル証明文とを出力する証明付再暗号シャッフル装置であって、

（ａ）前記入力文列から出力暗号文列を生成するとともに、前記入力文列から出力暗号文への変換情報の保有に関するコミットメント（「変換情報保有コミットメント」という）を生成する変換情報保有コミットメント生成処理と、（ｂ）前記変換の満たす条件に関するコミットメント（「変換条件コミットメント」という）を生成する、変換条件コミットメント生成処理と、（ｃ）再暗号シャッフル情報と挑戦値とから応答を生成する応答生成処理と、（ｄ）前記変換情報保有コミットメントと前記変換条件コミットメントと前記応答とを前記再暗号シャッフル証明文として出力する処理、の前記（ａ）乃至（ｄ）の処理を再暗号シャッフル装置を構成するコンピュータで実行させるためのプログラムを記録している。

【００２２】本発明に係る記録媒体は、入力文列と、前記証明付再暗号シャッフル装置から出力される前記出力暗号文列と、前記証明付再暗号シャッフル装置から出力される、前記入力文列から前記出力暗号文への変換情報の保有に関する変換情報保有コミットメントと、前記変換の満たす条件に関する変換条件コミットメントと、前記応答とからなる再暗号シャッフル証明文とを入力とし、受理または不受理である検証結果を出力する再暗号シャッフル検証装置であって、（ａ）前記入力文列と、前記出力暗号文列と、前記変換情報保有コミットメントと、応答と、挑戦値とより、前記入力文列から前記出力暗号文列への変換情報を保有していることを検証する、変換情報保有検証処理と、（ｂ）前記変換条件コミットメントと、前記応答と、前記挑戦値とから、前記入力文列から前記出力暗号文列への変換の満たす条件を検証する、変換条件検証処理と、（ｃ）前記変換情報保有検証処理と前記変換条件検証処理がともに受理された場合に、再暗号シャッフル検証結果として受理を出力し、それ以外は不受理を出力する処理、の前記（ａ）乃至（ｃ）の処理を再暗号シャッフル検証装置を構成するコンピュータで実行させるためのプログラムを記録している。

【００２３】［発明の概要］本発明は、再暗号シャッフルを、より一般的な変換の一種として表現し、この変換の情報を保有していることの証明と、この変換の満たす条件の証明との二つを合わせて、再暗号シャッフルの証明を構成している。

【００２４】これら二種の証明それぞれは、従来の再暗号シャッフルの証明よりも単純であり、入力暗号文数に依存せずに、証明の計算量が短縮されたものであり、また、これらを合わせた再暗号シャッフルの証明でも、この優位性が保たれる。

【００２５】変換の情報を保有していることの証明は、出力暗号文列と変換情報保有コミットメントを生成した後、挑戦値から上記変換と、変換情報保有コミットメント生成に使用する乱数とに依存して、応答を生成することにより行う。

【００２６】ここで、応答と挑戦値の関係に変換が反映されるため、変換の満たす条件から挑戦値に依存せずに、応答と挑戦値の満たす関係式が存在する。この関係式をコミットして変換の満たす条件を証明する。

【００２７】証明すべき変換の満たす条件として、再暗号シャッフルを表わす変換の満たす条件を選べば、両証明をもってして、再暗号シャッフルの証明を構成することができる。

【００２８】

【発明の実施の形態】本発明の上記および他の目的、特徴および利点を明確にすべく、以下添付した図面を参照しながら本発明の実施の形態につき詳細に説明する。

【００２９】最初に前提となる事柄について述べる。本発明において用いられる暗号方法は、確率暗号である公開鍵暗号系に属する暗号方法である。例えば、Elgamal暗号、楕円暗号、代数曲線暗号などもこれに含まれる。

【００３０】本発明に係る証明付再暗号シャッフル方法は、入力暗号文の作成者全員が証明付再暗号シャッフルを行う証明者に対して、入力暗号文の制作に用いた秘密変数を漏らさなければ、証明者は再暗号シャッフル証明文を偽造できないものである。ただし、本発明に係る入力文列生成方法を併せて用いることで、入力暗号文の作成者が証明者と共謀した場合においても、証明文の偽造を防止する、ことができる。

【００３１】本発明に係る証明付再暗号シャッフル方法は、変換情報保有コミットメントを生成する変換情報保有コミットメント生成処理と、変換条件コミットメントを生成する変換条件コミットメント生成処理と、応答や準応答を生成する応答生成処理とよりなり、証明文はこの上記３種類の処理より生成されるコミットメントと応答(応答および準応答)とよりなる。

【００３２】本発明に係る再暗号シャッフル検証方法

は、入力文列と出力暗号文列と変換情報保有コミットメントと応答とから変換の情報を保有していることを検証する変換情報保有検証処理と、変換条件コミットメントと応答と準応答とから変換の満たす条件を検証する変換条件検証処理とよりなる。

【００３３】［変換情報保有コミットメント生成処理］証明付再暗号シャッフル方法を構成する変換情報保有コミットメント生成処理について説明する。

【００３４】変換情報保有コミットメント生成処理は、入力文列から再暗号シャッフルに対応する変換を行って出力暗号文列を生成し、入力文列から乱数による一般の変換を行って変換情報保有コミットメントを生成する。

【００３５】また入力文列に入力暗号文列と公開鍵以外が含まれている場合には、それから再暗号シャッフルに対応する変換を行ったものも変換情報保有コミットメントとする。

【００３６】応答を複数個生成する場合には、異る乱数による一般の変換を、複数個行い、変換情報保有コミットメントを、複数組生成する。

【００３７】例えば、この変換を、出力暗号文列および変換情報保有コミットメントを、入力文列を基底として、再暗号化に用いた変数および乱数および並び替えに対応する値とを表現とした表現値として生成できる。

【００３８】また、この表現とは、基底とある表現値を対応付けるものであり、かつ、基底と表現値から表現を計算することが計算量的に困難となる方法である必要があり、この表現方法に、冪乗剰余を用いることができる。

【００３９】例えば、入力暗号文列を、$g[i,\Gamma];i=1,\ldots,n;\Gamma=0,\ldots,l$、公開鍵を、$g[i,\Gamma];i=n+1,\ldots,n+m;\Gamma=0,\ldots,l$、それ以外の入力文列の成分を、$g[i,\Gamma];i=1,\ldots,n+m;\Gamma=l+1,\ldots,l'$、一般の変換に対応する乱数(以下「情報隠蔽因子」と呼ぶ)を、$A[\mu,j];\mu=1,\ldots,n+m,j=n+1,\ldots,n+m'$、再暗号化の変数を、$A[i,j];i=n+1,\ldots,n+m,j=1,\ldots,n$、並び替えに対応する変換を表す変数を、$A[i,j];i,j=1,\ldots,n$として、出力暗号文列　$g''[i,\Gamma];i=1,\ldots,n;\Gamma=1,\ldots,l$を、

$$g''[i,\Gamma]=\Pi_{j=1}^{n}g[j,\Gamma]^{A[j,i]}\ \Pi_{j=n+1}^{n+m}g[j,\Gamma]^{A[j,i]}/F_p\quad i=1,\ldots,n\ \Gamma=1,\ldots,l$$

と生成し、変換情報保有コミットメントを、

$$g''[i,\Gamma]=\Pi_{j=1}^{n}g[j,\Gamma]^{A[j,i]}\ \Pi_{j=n+1}^{n+m}g[j,\Gamma]^{A[j,i]}/F_p\quad i=n+1,\ldots,n+m'\quad\Gamma=1,\ldots,l$$

と生成し、入力文列に$g[i,\Gamma];i=1,\ldots,n+m;\Gamma=l+1,\ldots,l'$が含まれた場合の変換情報保有コミットメントを、

$$g''[i,\Gamma]=\Pi_{j=1}^{n}g[j,\Gamma]^{A[j,i]}\ \Pi_{j=n+1}^{n+m}g[j,\Gamma]^{A[j,i]}/F_p\quad i=1,\ldots,n+m'\ \Gamma=l+1,\ldots,l'$$

と生成できる。

【００４０】また、これらをまとめて

$$g''[i,\Gamma]=\Pi_{j=1}^{n+m}g[j,\Gamma]^{A[j,i]}/F_p\quad i=1,\ldots,n+m'\quad\Gamma=1,\ldots,l'$$

と記述できる。

【００４１】この時、$g''[i,\Gamma];i=1,\ldots,n+m;\ \Gamma=1,\ldots,l$を「出力文列」と呼ぶ。ここで、$g''[\mu,\Gamma]$が表現値で、$A[\mu,\nu]$が表現で、$g[\mu,\Gamma]$が基底である。

【００４２】変換情報保有コミットメントを応答の数に応じて、複数組生成する場合には、異る$A[\mu,j];\mu=1,\ldots,n+m,j=n+1,\ldots,n+m'$を複数用意して生成する。

【００４３】この変換情報保有コミットメントと、入力文列と出力暗号文列および挑戦値に対応する応答を証明者が検証式を満たすように生成できることが、入力文列から出力暗号文列への変換の知識を有していることの証明となる。

【００４４】［変換条件コミットメント生成処理］証明付再暗号シャッフル方法を構成する変換条件コミットメント生成処理について説明する。

【００４５】応答と挑戦値の関係に、入力文列から出力文列および変換情報保有コミットメントへの変換の満たす条件が反映される。そのため、挑戦値に依存せずに成り立つ、応答と挑戦値の関係式が存在する。この関係式をこの変換の満たす条件を表現するものとしてコミットしたものを変換条件コミットメントとする。

【００４６】応答を複数個生成する場合は、知識の隠蔽因子の違いを関係式に反映させる。例えばこの関係式を、応答と挑戦値の多項式である恒等式とし、その係数をコミットするか、あるいはこの多項式の一部の項を準応答として、準応答の係数をコミットして変換条件コミットメントとできる。また挑戦値決定後に応答と準応答を生成すればよい。

【００４７】応答の各成分は挑戦値の多項式であるが、一部のこの多項式の、一部の項の各2乗の和と、挑戦値の成分の一部の各２乗の和とが、挑戦値によらずに等しくなる関係を内包しているものや、上記にて各３乗の和とが、挑戦値によらずに等しいくなる関係を内包している恒等式を実施例では用いている。

【００４８】これに対応する実施例における恒等式は、挑戦値$c[i]$、応答$r[i]$を用いて、

$$\Sigma_{i=1}^{n}(\Sigma_{j=1}^{n}A[i,j]c[j])^2=\Sigma_{i=1}^{n}c[i]^2/F_q$$

や、

$$\Sigma_{i=1}^{n}(\Sigma_{j=1}^{n}A[i,j]c[j])^3=\Sigma_{i=1}^{n}c[i]^3/F_q$$

の関係を内包したものを用いている。

【００４９】なお、

$$\Sigma_{j=1}^{n}A[i,j]c[j]/F_q\quad i=1,\ldots,n$$

は、$r[i]$を構成する挑戦値の多項式

$$\Sigma_{j=1}^{n+m}{}'A[i,j]c[j]/F_q\quad i=1,\ldots,n$$

の一部である。

【００５０】例えばこれらの関係式は、入力文列から出力暗号文列および変換情報保有コミットメントへの変換を定義している変数$A[\mu,\nu];\mu=0,\ldots,n+m;\nu=0,\ldots,n+m'$における$A[i,j];i,j=0,\ldots,n$が、正規直交行列であ

ることや、準置換行列であることの性質を反映した関係式である。

【００５１】「置換行列」とは、正方行列で、どの行どの列にもただ一つだけ0でない成分が存在し、その値が１である行列のことである。正規直交行列でありかつ準置換行列である行列は置換行列である。

【００５２】「準置換行列」とは、上記置換行列の1である成分を、１の３乗根のいずれかで置き換えたものとする。ただし、それぞれの成分毎に異なる１の３乗根で置き換えを行っても良い。この時、置換行列に対応する＊10

$$\Sigma_{i=1}^{n} r[i]r[i]r[i]+\rho''r'+\Sigma_{\mu=1}^{n+m} \rho'[\mu]r[\mu] /F_q$$
$$=\Sigma_{i=1}^{n} r[i]r[i]r[i]+\rho''(\lambda[0]+\Sigma_{i=1}^{n} \lambda[i]r[i]r[i])+\Sigma_{\mu=1}^{n+m} \rho'[\mu]r[\mu] /F_q$$
$$=\Sigma_{i=1}^{n} c[i]c[i]c[i]+\Sigma_{i=1}^{n} \psi[i]c[i]c[i]+\Sigma_{\mu=1}^{n+m} '\phi[\mu]c[\mu] /F_q$$

があげられる。

【００５４】ここでは、変換の満たす条件に対応する関係式を内包するように恒等式の係数ρ'', $\rho'[i]$, $\phi[\mu]$, $\psi[i]$を求めなければならない。

【００５５】また恒等式の一部

$$r'=\lambda[0]+\Sigma_{i=1}^{n} \lambda[i]r[i]r[i] /F_q$$

を準応答として、準元係数$\lambda[\mu];\mu=0,\ldots,n$をコミットする場合もある。

【００５６】変換条件コミットメントとして、恒等式の係数または、それをコミットしたものと、準元係数をコミットしたものを生成する。実施例では恒等式の一部を、

$$v, v^{\phi[0]} /F_p$$

のようにコミットし、準元係数を

$$u, u^{\lambda[\mu]} /F_p \quad \mu=0,\ldots,n$$

のようにコミットする。

【００５７】恒等式の係数をコミットすることと、準応答を用いることには、検証者が応答とコミットメントから再暗号シャッフルを特定するための情報を減じる効果がある。

【００５８】［応答生成処理］証明付再暗号シャッフル方法を構成する応答生成処理について説明する。

【００５９】応答生成処理では、変換情報保有コミットメントと変換条件コミットメントと入力文列と出力暗号文列を挑戦値生成関数に入力して挑戦値をえる。

【００６０】ここで、「挑戦値生成関数」とは、出力から入力を求めることや、異る出力成分間の関係を意図して入力を決定することが計算量的に困難である関数である。これにより挑戦値が入力とコミットメントと出力とが決定後に、証明者の意図を入れずに生成されたことが保証できる。

【００６１】挑戦値生成関数を用いない場合には、検証者が、入力と出力とコミットメントとが示された後に無作為に選ぶことで挑戦値を得る。

【００６２】挑戦値から、再暗号シャッフル方法と情報隠蔽因子とを反映した応答や準応答を生成する。

＊の変換は再暗号シャッフルに対応している。すなわち、この変換条件コミットメント生成処理により変換の満たす条件を証明することによって変換が再暗号シャッフルであることを証明できる効果がある。

【００６３】例えば、上記関係式を内包した恒等式の例として、

$$\Sigma_{i=1}^{n} r[i]r[i]+\Sigma_{\mu=1}^{n+m} \rho'[\mu]r[\mu] /F_q$$
$$=\Sigma_{i=1}^{n} c[i]c[i]+\Sigma_{\mu=1}^{n+m} '\phi[\mu]c[\mu] /F_q$$

や、

$$\Sigma_{i=1}^{n} r[i]r[i]r[i]+\rho''r'+\Sigma_{\mu=1}^{n+m} \rho'[\mu]r[\mu] /F_q$$
$$=\Sigma_{i=1}^{n} r[i]r[i]r[i]+\rho''(\lambda[0]+\Sigma_{i=1}^{n} \lambda[i]r[i]r[i])+\Sigma_{\mu=1}^{n+m} \rho'[\mu]r[\mu] /F_q$$
$$=\Sigma_{i=1}^{n} c[i]c[i]c[i]+\Sigma_{i=1}^{n} \psi[i]c[i]c[i]+\Sigma_{\mu=1}^{n+m} '\phi[\mu]c[\mu] /F_q$$

【００６３】応答や準応答を複数個生成する場合は、各応答は異る情報隠蔽因子を反映させる。

【００６４】例えば、出力暗号文列および変換情報保有コミットメントを基底として挑戦値を表現とする表現値と、入力文列を基底として応答を表現とする表現値が等しくなるような応答を生成すればよい。

【００６５】例えば、応答$r[\mu];\mu=1,n+m$は、挑戦値$c[\mu];\mu=1,\ldots,n+m'$を用いて、

$$r[\mu]=\Sigma_{\nu=1}^{n+m} 'A[\mu,\nu]c[\nu] /F_q \quad \mu=1,\ldots,n+m$$

を、準応答として、

$$r'=\lambda[0]+\Sigma_{i=1}^{n} \lambda[i]r[i]r[i] /F_q$$

を生成する。

【００６６】［変換情報保有検証処理］再暗号シャッフル検証方法を構成する変換情報保有検証処理について説明する。

【００６７】入力文列と出力暗号文列と変換情報保有コミットメント間の関係を、応答と挑戦値の関係が反映していることを検証する。例えば、出力暗号文列および変換情報保有コミットメントを基底として挑戦値を表現とする表現値と、入力文列を基底として応答を表現とする表現値が等しくなるような応答と挑戦値の関係が存在することを確認する。

【００６８】例えば、挑戦値$c[i];i=1,\ldots,n+m'$と応答$r[i];i=1,\ldots,n+m$が、

$$\Pi_{i=1}^{n+m} 'g''[i,\Gamma]^{c[i]} =\Pi_{i=1}^{n+m} g[i,\Gamma]^{r[i]} /F_p$$
$$\Gamma=1,\ldots,l'$$

が成り立つことを確認する。

【００６９】挑戦値は、証明文生成に用いた値と同じ値を用いる。これは、挑戦値生成関数を用いる場合は、挑戦値生成関数への入力が証明文と入出力に存在するので可能である。

【００７０】［変換条件検証処理］再暗号シャッフル検証方法を構成する変換条件検証処理について説明する。

【００７１】変換条件コミットメントより、挑戦値と応答が変換の満たす条件を反映した関係を満たしていることを検証する。

【００７２】例えば、変換の満たす条件を反映した関係を内包する恒等式に、応答と挑戦値または、応答と挑戦値と準応答を代入して、恒等式が成立することを確認している。また準応答がある場合、応答と準応答と準元係数をコミットしたものより、準応答の正当性も確認する。

【００７３】例えば、変換条件コミットメントとして恒＊

$$\Sigma_{i=1}^{n} r[i]r[i]r[i]+\rho''r'+\Sigma_{\mu=1}^{n+m}\rho'[\mu]r[\mu]\ /F_q$$
$$=\Sigma_{i=1}^{n} r[i]r[i]r[i]+\rho''(\lambda[0]+\Sigma_{i=1}^{n}\lambda[i]r[i]r[i])+\Sigma_{\mu=1}^{n+m}\rho'[\mu]r[\mu]\ /F_q$$
$$=\Sigma_{i=1}^{n} c[i]c[i]c[i]+\Sigma_{i=1}^{n}\psi[i]c[i]c[i]+\Sigma_{\mu=1}^{n+m}{}'\phi[\mu]c[\mu]\ /F_q$$

が成り立つことを確認し、また準応答の正当性を、検証式

$$u^{r'}=u[0]\prod_{i=1}^{n}u[i]^{r[i]r[i]}\ /F_p$$

が成り立つことより確認する。

【００７４】また恒等式の係数の一部がコミットされて※

$$v^{\wedge}\{\Sigma_{i=1}^{n} r[i]r[i]r[i]+\rho''r'+\Sigma_{\mu=1}^{n+m}\rho'[\mu]r[\mu]\}\ /F_p$$
$$=v^{\wedge}\{\Sigma_{i=1}^{n} r[i]r[i]r[i]+\rho''(\lambda[0]+\Sigma_{i=1}^{n}\lambda[i]r[i]r[i])+\Sigma_{\mu=1}^{n+m}\rho'[\mu]r[\mu]\}\ /F_p$$
$$=v^{\wedge}\{\Sigma_{i=1}^{n} c[i]c[i]c[i]+\Sigma_{i=1}^{n}\psi[i]c[i]c[i]+\Sigma_{\mu=1}^{n+m}{}'\phi[\mu]c[\mu]\}\ /F_p$$

が成り立つことを確認する。上式で、記号「^」は指数演算を示す。

【００７５】［入力文列生成方法］本発明に係る証明付再暗号シャッフル方法は、応答と挑戦値の関係に、入力文列から出力暗号文列と変換情報保有コミットメントへの変換を反映する必要があった。そのためには、挑戦値が与えられたときに生成できる応答が制限されていなければならない。しかし、入力暗号文の生成情報を証明者が知っている場合この制限を破れる可能性がある。そしてこれを阻止するための方法が入力文列生成方法である。

【００７６】本発明に係る入力文列生成方法は、疑似乱数を生成し、これにより入力文列を変換するか、この疑似乱数を入力文列に加えることにより、入力暗号文の生成者にさえ入力文列を決定できない入力文列を生成する。

【００７７】［入力文列生成方法(1)］疑似乱数を生成して、入力暗号文列と公開鍵にくわえて入力文列とする。この時疑似乱数を決められた入力より決定して再現性を保証する。

【００７８】例えば、入力暗号文列を、$g[i,\Gamma];i=1,\ldots,n;\Gamma=0,\ldots,l$、公開鍵を$g[i,\Gamma];i=n+1,\ldots,n+m;\Gamma=0,\ldots,l$としたとき、決まった入力から疑似乱数を、$(n+m)\times(l'-l)$;$l'-l\geqq 1$個生成し、これを、$g[i,\Gamma];i=1,\ldots,n+m;\Gamma=l+1,\ldots,l'$として、入力文列を、
$g[i,\Gamma]i=1,\ldots,n+m;\Gamma=1,\ldots,l'$
とする。

【００７９】［入力文列生成方法(2)］入力暗号文列を構成する各暗号文と公開鍵を、入力暗号文列と公開鍵を入力として生成した公開鍵列を構成する各公開鍵で再暗＊等式の係数$\rho'',\rho'[\mu],\phi[\mu],\psi[i]$に対して、挑戦値$c[i];i=1,\ldots,n+m'$と、応答$r[i];i=1,\ldots,n+m$が、恒等式

$$\Sigma_{i=1}^{n} r[i]r[i]+\Sigma_{\mu=1}^{n+m}\rho'[\mu]r[\mu]=\Sigma_{i=1}^{n} c[i]c[i]+\Sigma_{\mu=1}^{n+m}{}'\phi[\mu]c[\mu]\ /F_q$$

や、恒等式

$$v^{\wedge}\{\Sigma_{i=1}^{n} r[i]r[i]+\Sigma_{\mu=1}^{n+m}\rho'[\mu]r[\mu]\}\ /F_p$$
$$=v^{\wedge}\{\Sigma_{i=1}^{n} c[i]c[i]+\Sigma_{\mu=1}^{n+m}{}'\phi[\mu]c[\mu]\}\ /F_p$$

や、

※いる場合は、代わりに、

号化して、これをあわせて入力文列とする。

【００８０】ここで「公開鍵列」とは、入力から疑似乱数を公開鍵列を構成する公開鍵の数と同数一意的に生成し、この各乱数が各公開鍵のいずれかの要素となるようにしたものをいう。

【００８１】例えば、公開鍵列を、$g'[i,\Gamma];i=1,\ldots,n+m;\Gamma=1,\ldots,l$とし、入力暗号文列を、$\eta[i,\Gamma];i=1,\ldots,n;\Gamma=0,\ldots,l$、公開鍵を、$\eta[i,\Gamma];i=n+1,\ldots,n+m;\Gamma=0,\ldots,l$としたとき、入力文列 $g[i,\Gamma];i=1,\ldots,n+m;\Gamma=l+1,\ldots,l$を、検証者にも明らかな任意の正整数$s[i];i=1,\ldots,n+m$を用いて、

$$g[i,\Gamma]=\eta[i,\Gamma]g'[i,\Gamma]^{s[i]}\ /F_p$$

と表す。$s[i]$として、例えば$s[n+m]=0,s[j]=1;j=1,\ldots,n+m-1$を選ぶ。

【００８２】［入力文列生成方法(3)］各入力平文を公開鍵列を構成する対応する各公開鍵を用いて暗号化し、この公開鍵を用いて暗号化したことの証明をする。この証明文が受理された暗号文と、公開鍵をあわせて入力文列とする。

【００８３】例えば、公開鍵列を、$g'[i,\Gamma];i=1,\ldots,n+m;\Gamma=0,1$とし、平文を、$m[i];i=1,\ldots,n;\Gamma=0,1$としたとき、入力暗号文を、

$$\eta[i,0]=g'[i,0]^{s[i]}\ /F_p\quad i=1,\ldots,n$$
$$\eta[i,1]=m[i]g'[i,1]^{s[i]}\ /F_p\quad i=1,\ldots,n$$

と生成し、合わせて、$\eta[i,0]=g'[i,0]^{s[i]}\ /F_p$ となる$s[i]$の知識を証明することで、$g'[i,0]$を使って暗号化した証明文とする。

【００８４】この証明文が検証された暗号文より、入力文列を

$g[i,\Gamma]=\eta[i,\Gamma]\quad i=1,\ldots,n;\Gamma=0,1$

g[i,Γ]=g'[i,Γ]　i=n+1,...,n+m;Γ=0,1
とする。

【００８５】［証明付公開鍵列生成方法］与えられた入力から一意的に疑似乱数列を発生させ、その各乱数から与えられた手続きにより作られた値を成分に持ち、かつ同じ秘密鍵を持つ公開鍵を、各乱数に対応して複数個生成する。これと同時に、全ての公開鍵が同じ秘密鍵を持つことの証明文を生成する。

【００８６】複数人で秘密鍵を分散所持している場合は、各人が各秘密鍵で分散して公開鍵列を生成した後、それらを合わせて統合した公開鍵列を生成する。

【００８７】例えば、疑似乱数生成器Hash(*)が与えられ、入力*から出力を得る。そして出力をまた入力することを繰り返すことにより、再帰的に、疑似乱数列 を生成する。この数列の各値をk乗してできた数列から0,1を除いて得られるn+m個の値よりなる数列g'[i,0];i=1,...,n+mの各値を成分に持ち、同じ秘密鍵を持つ公開鍵g'[i,Γ];i=1,...,n+m;Γ=0,...,lを各乱数に対応して生成する。

【００８８】ここで秘密鍵をx[Γ];Γ=1,...,lとすると、公開鍵列は、
g'[i,0]=g'[i,0]
$g'[i,\Gamma]=g'[i,0]^{x[\Gamma]}/F_p$　i=1,...,n+m;Γ=1,...,l
と表せる。

【００８９】上記公開鍵列を正しく生成したことの証明文を生成する。

【００９０】秘密鍵を分散所持している場合、各自で分散秘密鍵に対応する公開鍵列を生成し、最後にそれらを合わせて秘密鍵に対する公開鍵列を生成する。

【００９１】
【実施例】本発明の実施例について図面を参照して説明する。以下の実施例では、Elgamal暗号を使った例に即して説明する。図中では、略語が使用されており、例えば保有コミットとは変換情報保有コミットメント、条件コミットとは変換条件コミットメント、恒等式コミットとは恒等式係数のコミットメント、準応答コミットとは準応答の係数のコミットメント、保有処理とは変換情報保有コミットメント生成処理、条件処理とは変換条件コミットメント生成処理、応答処理とは応答生成処理、保有検証処理とは変換情報保有検証処理、条件検証処理とは変換条件検証処理のことをいう。

【００９２】図３は、本発明に係る証明付再暗号シャッフル装置および再暗号シャッフル検証装置の実施例における入出力について示す図である。

【００９３】図３を参照すると、本発明の一実施例においては、複数の入力暗号文322と公開鍵323からなる入力文列300と、並び替え方を決めるシャッフル行列307と、再暗号化の変数である再暗号秘密乱数305と、変換情報保有コミットメントを生成するための乱数である情報隠蔽因子306と、からなる再暗号シャッフル行列304と、恒等式の係数の種となる元係数308と、恒等式の一部である準応答319の係数である準元係数309と、これらをコミットするための係数基底310とよりなる変換条件コミットメントを生成するための諸定数と、を含む再暗号シャッフル情報303と、が、証明付再暗号シャッフル装置312に入力され、出力暗号文列313と再暗号シャッフル証明文314が出力される。

【００９４】再暗号シャッフル証明文314は、恒等式の係数、または、それをコミットしたものと準応答の係数をコミットしたものとを含む変換条件コミットメント316と、変換情報保有コミットメント315と、応答317と、準応答318とを含む。

【００９５】これら入力文列300と、出力暗号文列313と、再暗号シャッフル証明文314とが、再暗号シャッフル検証装置319に入力され、受理または不受理の検証結果320が出力される。

【００９６】上記証明付再暗号シャッフル装置は、証明者が入力暗号文列の生成情報を知らない時のみに、再暗号シャッフル証明文を偽造できない。この偽造をいかなる場合にも阻止するために加える方法が、入力文列生成方法であり、３種類の入力文列生成方法の実施例を挙げる。また、このうち二つの入力文列生成方法で使用される証明付公開鍵列生成方法についても説明する。

【００９７】以下では、証明付再暗号シャッフル方法と、入力文列生成方法と、証明付個別公開鍵列生成方法に共通して、前提となる事項から順に説明していく。

【００９８】［Elgamal領域変数］まずElgamal領域変数について説明する。

【００９９】この変数は、二つの素数p,qであり、これらは、p=kq+1なる関係を満たす。ここで、kは整数である。

【０１００】［挑戦値生成関数と基底生成関数］挑戦値生成関数と基底生成関数について説明する。これらは順に、
Hash[μ;μ=0,...,n](*),Hash'[μ;μ=0,...,n](*)
とする。

【０１０１】両関数の添字についているギリシャ文字μは0からnまでの値をとり、引数「*」を入力すると、それぞれn+1成分のベクトルを出力する。

【０１０２】挑戦値生成関数の出力は、n+1個の1,0でないq以下の整数、基底生成関数の出力は、n+1個の1,0でないp以下の整数で位数qのF_pの元(位数p-1の乗法群の位数q部分群の元)である整数である。

【０１０３】また、これらの関数は入出力間や出力の異なる成分間の関係を計算量的に意図して引数を決定できない関数とする。

【０１０４】基底生成関数の具体的な構成方法の例としては、|p|ビットを出力するハッシュ　関数Hash(*)を一つ用意して、
Hash(*)

を計算し、次に、この計算結果をハッシュ関数の引数に入力してさらに計算結果を得る。これを、繰り返すことにより、再帰的に、数列h[0],h[1],h[2],...を生成し、その各数値をk乗をした数列$h[0]^k$,$h[1]^k$,$h[2]^k$,...を求める。この中から順に、1,0でないものをn+1個選んでいく。

【０１０５】挑戦値生成関数の場合は、|q|ビットを出力するハッシュ関数を用いて数列を求め、その中から順に1,0でないものをn+1個選んでいく(この場合、k乗する操作は必要無い)。

［公開鍵］公開鍵について説明する。公開鍵は、二つの数値$\eta[0,0]$, $\eta[0,1]$であり$\eta[0,0]$は位数qのF_pの元とする。$\eta[0,1]$は秘密鍵xを用いて、

$\eta[0,1]=\eta[0,0]^x$ /F_p

と計算される。

【０１０６】［入力暗号文］入力暗号文について説明する。平文を、p以下で位数qのF_pの元から選び、これをMとする。これから疑似乱数生成器で生成した秘密乱数rを用いて、入力暗号文を、

$(\eta[0,0]^r, M\ \eta[0,1]^r)$ /F_p

と計算する。

【０１０７】［再暗号化］再暗号化について説明する。Elgamal暗号文$(\eta[0,0]^r, M\ \eta[0,1]^r)$ /F_pが与えられた時、任意の乱数sを選んで、

$(\eta[0,0]^r, M\eta[0,1]^r)\rightarrow(\eta[0,0]^r\ \eta[0,0]^s, M\eta[0,1]^r\ \eta[0,1]^s)$ /F_p=$(\eta[0,0]^{r+s}, M\eta[0,1]^{r+s})$ /F_p

なる変換を行うことを「再暗号化」という。上記変換はrを知らなくても実行できる。またこの変換により再暗号化された暗号文の復号結果はかわらない。この時の乱数sを、「再暗号秘密乱数」と呼ぶ。

【０１０８】［置換行列］置換行列について説明する。「置換行列」とはどの行にもまたどの列に対しても０でない成分が唯一存在し、１の値をとる。ただし本実施例ではFq上で考える。下に例をあげる。

【０１０９】

０００１０
１００００
０１０００
０００１
００１００ /F_q

【０１１０】［準置換行列］準置換行列ついて説明する。「準置換行列」とは、置換行列の1である成分を、F_p上の３個ある１の３乗根のいずれかで置き換えたものと定義する。これらをw,w^2,1として下に準置換行列の例をあげる。

【０１１１】

０００$w^2$０
w００００
０$w^2$０００
０００１
００w００ /F_q

【０１１２】［再暗号シャッフル］再暗号シャッフルについて説明する。入力暗号文列$\eta[i,0]$, $\eta[i,1]$;i=1,...,nの順序を入れ替えて、暗号文列$\eta'[i,0]$, $\eta'[i,1]$;i=1,...,nを生成し、さらにn個の秘密乱数s[i]:i=1,...,nと、公開鍵$\eta[0,0]$, $\eta[0,1]$を用いて、出力暗号文列$g''[i,\Gamma]$;i=1,...,n, Γ=0,1を、

$g''[i,\Gamma]=\eta'[i,\Gamma]\,\eta[0,\Gamma]^{s[i]}$ /F_p i=1,...,n, Γ=0,1

と計算する。これが、再暗号シャッフルの出力結果である。これを、「出力暗号文列」と呼ぶ。

【０１１３】［再暗号シャッフル行列］再暗号シャッフル行列について説明する。「再暗号シャッフル行列」とは、n+1行n+1列の行列で、その成分A[μ, ν];μ, ν=0,...,nが、

A[μ, ν]=
A[i,j] i,j=1,...,n シャッフル行列307
A[0,j]$\in_R$ j=1,...,n 再暗号秘密乱数305
A[i,0]$\in_R$ i=1,...,n 情報隠蔽因子306
A[0,0]$\in_R$ 情報隠蔽因子306

であるものである。

【０１１４】［再暗号シャッフル行列変換］再暗号シャッフル行列変換について説明する。これは、入力文列g[μ, Γ]に以下のように作用して、出力文列$g''[\mu,\Gamma]$を出力する。

【０１１５】$g''[\mu,\Gamma]=\Pi_{\nu=0}{}^{n} g[\nu,\Gamma]A[\nu,\mu]$ /F_p
μ=0,...,n, Γ=0,1

ここで、シャッフル行列が置換行列の場合、出力暗号文列を、g''[i,0],g''[i,1];i=1,...,nとし、展開すると、ある置換(i,j|π(i)=j)に対して、

$g''[j,0]=g[i,0]\,\eta[0,0]^{A[0,j]}$ /F_p
$g''[j,1]=g[i,1]\,\eta[0,1]^{A[0,j]}$ /F_p

となり、これは再暗号シャッフルの出力となる。

【０１１６】またシャッフル行列が準置換行列の場合、準再暗号シャッフルの結果

$g''[j,0]=g[i,0]^{w[i]}\ \eta[0,0]^{A[0,j]}$ /F_p
$g''[j,1]=g[i,1]^{w[i]}\ \eta[0,1]^{A[0,j]}$ /F_p

を出力する(準再暗号シャッフルとは各出力暗号文を1またはwまたはw^2乗すると再暗号シャッフルとなるものと定義する)。ここでw[i];i=1,...,nはFq上の1の三乗根のいずれかをとる。

【０１１７】［実施例(1)］本発明の一実施例をなす証明付再暗号シャッフル方法およびその検証方法について、図４、図５を参照して説明する。以下で、Γ=0,1を取るものとする。

【０１１８】再暗号シャッフル情報401として、再暗号シャッフル行列402、係数基底404、元係数403を以下のように準備する。

【０１１９】再暗号シャッフル行列402に関しては、まず１からnまでの数を順に並べる。疑似乱数発生器（不

図示）をn回使って、n個数列を発生させ、そのi番目の数をn-i+1で割り余りの数を求めπ'(i)とする。

【０１２０】iは1からn迄順に、上記並べた数の下からπ'(i)番目の数をπ(i)とし、上記数列からこの数を取り除く作業を行いπ(i);i=1,...,nを決定する。シャッフル行列の第i行目はπ(i)列目の成分のみ値を1とし、その他を0とする。以上のようにして置換行列を生成する。

【０１２１】再暗号シャッフル行列のシャッフル行列以外の成分を以下のようにして生成する。まず、疑似乱数発生器により2n+1個のF_q上の数を作成し、A[i,0],A[0,j],A[0,0];i,j=1,...,nに割り振る。以上を合わせて再暗号シャッフル行列とする。

【０１２２】係数基底404　ν、元係数403 r'[0]を生成に関しては、疑似乱数生成器で1,0でないF_q上の数を生成し、r'[0]とし、疑似乱数生成器により/F_pの元を生成しF_p上でそのk乗をとり1,0でないものを選び位数qのF_pの元を生成し、νとする。

【０１２３】$r'[0]\in_R Z_q$,≠0,1
$v\in_R Z_p$,≠0,1 ,s.t. $v^q=1$ /F_p
入力暗号文列η[i,0],η[i,1];i=1,...,nと、公開鍵η[0,0],η[0,1]より、入力文列400
g[μ,Γ];μ=0,...,n;Γ=0,1を、
$g[0,\Gamma]=\eta[0,\Gamma]$　　Γ=0,1
$g[i,\Gamma]=\eta[i,\Gamma]$ /F_p　i=1,...,n, Γ=0,1
とする。

【０１２４】以下、証明付再暗号シャッフル方法を使う。

【０１２５】変換情報保有コミットメント生成処理419における再暗号シャッフル行列作用405により、上記暗号シャッフル行列402を入力文列400に以下の様に作用させて、出力文列406　g''[μ,Γ];μ=0,...,n;Γ=0,1を、
$g''[\mu,\Gamma]=\Pi_{\nu=0}^{n} g[\nu,\Gamma]^{A[\nu,\mu]}$ /F_p　μ=0,...,n, Γ=0,1
と生成する。

【０１２６】ここで、g''[i,Γ];i=1,...,n;Γ=0,1を出力暗号文列407、g''[0,Γ];Γ=0,1を変換情報保有コミットメント408とする。

【０１２７】変換条件コミットメント生成処理420における恒等式係数計算409により、元係数403　r'[0]と、再暗号シャッフル行列402と、を用いて、恒等式係数410　φ[μ],r'[0]を、
r'[0]=r'[0]
$\phi[0]=\Sigma_{j=1}^{n} A[j,0]A[j,0]+r'[0]A[0,0]$ /F_q
$\phi[i]=2\Sigma_{j=1}^{n} A[j,0]A[j,i]+r'[0]A[0,i]$ /F_q　i=1,...,n
と生成する。

【０１２８】さらに、係数基底404　νを用いて、隠蔽処理411により、恒等式係数410　r'[0],φ[0]を、
$v'=v^{r'^{[0]}}$ /F_p
$\omega=v^{\phi^{[0]}}$ /F_p
とコミットする。

【０１２９】以上より、φ[i],ω,v',νを変換条件コミットメント412とする。

【０１３０】ここで、コミットメント40 Aを、変換情報保有コミットメント408と変換条件コミットメント412とする。

【０１３１】応答生成処理421により、以上の入力文列400と、出力暗号文列417と、コミットメント409と、を挑戦値生成関数413の引数として、挑戦値414を、
c[0]=1,
c[i]=Hash[i](g[ν,0],g[ν,1],g''[ν,0],g''[ν,1],ν,φ[ν],ω,v';ν=0,...,n)　i=1,...,n
と生成し、この挑戦値から再暗号シャッフル行列02を用いて、応答416を
$r[\mu]=\Sigma_{\nu=0}^{n} A[\mu,\nu]c[\nu]$ /F_q　μ=0,...,n
と生成415する。

【０１３２】以上のコミットメント40 Aと応答416を、再暗号シャッフル証明文418として出力し、再暗号シャッフルの結果として出力暗号文列417を出力する。

【０１３３】検証方法について、図５を参照して説明する。

【０１３４】再暗号シャッフル検証方法により、入力文列400　g[μ,Γ];μ=0,...,n;Γ=0,1と、出力暗号文列417　g''[i,Γ];i=1,...,n;Γ=0,1と、再暗号シャッフル証明文418中のコミットメント409である変換情報保有コミットメント408　g''[0,Γ];Γ=0,1と、変換条件コミットメント412　φ[ν],ω,v',ν;ν=0,...,n;Γ=0,1と、を挑戦値生成関数500に代入して、挑戦値501を、
c[0]=1
c[i]=Hash[i](g[ν,Γ],g''[ν,Γ],φ[ν],ω,v',ν;ν=0,...,n;Γ=0,1)i=1,...,n
と生成する。

【０１３５】変換情報保有検証処理505により、この挑戦値501を用いて入力文列400と、変換情報保有コミットメント408と出力暗号文列417である出力文列406と応答416とを用いて検証式、
$\Pi_{\mu=0}^{n} g[\mu,\Gamma]^{r[\mu]}=\Pi_{\mu=0}^{n} g''[\mu,\Gamma]^{c[\mu]}$ /F_p
Γ=0,1
が成り立つことを確認502する。

【０１３６】変換条件検証処理506により、挑戦値501と応答416と変換条件コミットメント412とを用いて検証式
$v'^{r[0]}\ v^{\wedge}\{\Sigma_{i=1}^{n} r[i]r[i]\}=\omega v^{\wedge}\{\Sigma_{i=1}^{n}(c[i]c[i]+\phi[i]c[i])\}$ /F_p
が成り立つことを確認503する。

【０１３７】以上全ての検証式が成り立てば、証明文を受理504する。

【０１３８】上記証明付再暗号シャッフル方法は、入力文列に対する再暗号シャッフル行列変換が少なくとも正

規直交行列に属するシャッフル行列を持つ再暗号シャッフル行列により行われたことを保証する効果がある。

【０１３９】入力暗号文と出力暗号文に制限が課せられており、この効果で、再暗号シャッフルの正当性を保証できる場合には、本実施例により、証明付再暗号シャッフルを構成できる。

【０１４０】例えば、入力暗号文は限られた候補から選ばれていることが証明されていて、かつそれらの候補は互いに他を基底として表現できないとする。この入力暗号文を再暗号シャッフルしたのち、復号していずれの復号文も正しい候補から選ばれたものであったとき、本実施例による証明文からこの再暗号シャッフルが正当であることが言える。

【０１４１】なお図４における変換情報保有コミットメント処理419、変換条件コミットメント生成処理420、応答生成処理421は、コンピュータ上で実行されるプログラムによりその処理・機能が実現される。また図５における変換情報保有検証処理505、変換条件検証処理506は、コンピュータ上で実行されるプログラムによりその処理・機能が実現される。この場合、該プログラムを記録した記録媒体（例えばＣＤ－ＲＯＭ、ＤＶＤ（digital versatile disk）、フロッピディスク媒体、ハードディスク媒体、磁気テープ媒体、半導体メモリ等のいずれか）から、該プログラムをコンピュータの主記憶にロードして実行することで、本発明を実施することができる。

【０１４２】［実施例(2)］本発明の実施例(2)の証明付再暗号シャッフル方法およびその検証方法について、図６、図７を参照して説明する。以下では、$\Gamma=0,1$を取るものとする。

【０１４３】再暗号シャッフル情報601として、再暗号シャッフル行列602、元係数603、係数基底604、605、準元係数606を、以下のように準備する。

【０１４４】まず再暗号シャッフル行列602に関しては、前記実施例(1)と同様に生成する。

【０１４５】元係数603　ρ',ρ''、係数基底604　v、係数基底605　u、準元係数606　$\lambda[\mu];\mu=0,\ldots,n$に関しても、実施例(1)と同様な手法で、$\rho',\rho''\ \lambda[\mu];\mu=0,\ldots,n$には1,0でない$F_q$上の数を、係数基底u,vには位数qの$F_p$の元を生成する。

【０１４６】$\rho'\in_R Z_q\ ,\neq 0,1$

$\rho''\in_R Z_q\ ,\neq 0,1$

$v\in_R Z_p\ ,\neq 0,1,\ s.t.\ v^q=1\ /F_p$

$\lambda[\mu]\in_R Z_q\ ,\neq 0,1,\ \mu=0,\ldots,n$

$u\in_R Z_p\neq 0\ ,1,\ s.t\ .u^q=1\ /F_p$

【０１４７】実施例(1)と同様にして、入力暗号文列と公開鍵より、入力文列600　$g[\mu\Gamma];\mu=0,\ldots,n;\Gamma=0,1$を生成する。

【０１４８】以下、証明付再暗号シャッフル方法を用いる。

【０１４９】実施例(1)と同様に、変換情報保有コミットメント生成処理623を行い、出力文列603　$g''[\mu,\Gamma];\mu=0,\ldots,n;\Gamma=0,1$を、生成する。ここで、$g''[i,\Gamma];i=1,\ldots,n;\Gamma=0,1$を、出力暗号文列604、$g''[0,\Gamma];\Gamma=0,1$を変換情報保有コミットメント605とする。

【０１５０】変換条件コミットメント生成処理625における恒等式係数計算606により、元係数603　ρ',ρ''、と、再暗号シャッフル行列602を用いて恒等式係数607　$\psi[i],\phi[i],\phi[0],\rho',\rho'';i=1,\ldots,n$を、

$\rho'=\rho'$

$\rho''=\rho''$

$\psi[i]=\Sigma_{j=1}^{n}(3A[j,0]+\rho''\lambda[j])A[j,i]\ /F_q\quad i=1,\ldots,n$

$\phi[i]=\Sigma_{j=1}^{n}(3A[j,0]A[j,0]A[j,i]+2\rho''\lambda[j]A[j,0]A[j,i])+\rho'A[0,i]\ /F_q\ i=1,\ldots,n$

$\phi[0]=\Sigma_{j=1}^{n}(A[j,0]A[j,0]A[j,0]+\rho''\lambda[j]A[j,0]A[j,0])+\rho''\lambda[0]+\rho'A[0,0]\ /F_q$

と生成する。

【０１５１】さらに、係数基底604　vを用いて、隠蔽処理608により、恒等式係数607　$\rho',\rho'',\phi[0]$を、

$\omega=v^{\phi[0]}\ /F_p$

$v''=v^{\rho''}\ /F_p$

$v'=v^{\rho'}\ /F_p$

とコミット609する。さらに、係数基底605　uを用いて、準元係数606　$\lambda[\mu];\mu=0,\ldots,n$を、

$u[0]=u^{\lambda[0]}\ /F_q$

$u[i]=u^{\lambda[i]}\ /F_q\quad i=1,\ldots,n$

とコミット612する。

【０１５２】以上より、$\psi[i],\phi[i],\omega,v'',v',v,u,u[0],u[i];i=1,\ldots,n$を変換条件コミットメント613とする。

【０１５３】ここで、コミットメント614を、変換情報保有コミットメント605と、変換条件コミットメント613とする。

【０１５４】応答生成処理624により、以上の入力文列600と、出力暗号文列604と、コミットメント614と、を挑戦値生成関数615の引数として、挑戦値616を、

$c[0]=1$

$c[i]=Hash[i](g[\nu,\Gamma],g''[\nu,\Gamma],u,u[\nu],v,\phi[j],\psi[j],\omega,v',v'';\Gamma=0,1,2;\nu=0,\ldots,n;j=1,\ldots,n)\quad i=1,\ldots,n$

と生成し、この挑戦値616から、再暗号シャッフル行列602を用いて、応答618を、

$r[\mu]=\Sigma_{\nu=0}^{n}A[\mu,\nu]c[\nu]\ /F_q\quad \mu=0,\ldots,n$

と生成617する。

【０１５５】さらに、準応答620を、準元係数606　$\lambda[\mu];\mu=0,\ldots,n$と、応答618より、

$r'=\lambda[0]+\Sigma_{i=1}^{n}\lambda[i]r[i]r[i]\ /F_q$

と生成619する。

【０１５６】以上のコミットメント614と、応答618と、

準応答620と、を再暗号シャッフル証明文622として出力し、再暗号シャッフルの結果として、出力暗号文列604を出力する。　検証方法について、図６及び図７を参照して、以下に説明する。

【０１５７】再暗号シャッフル検証方法により、入力文列600　$g[\mu,\Gamma];\mu=0,\ldots,n;\Gamma=0,1$と、出力暗号文列604　$g''[i,\Gamma];i=1,\ldots,n;\Gamma=0,1$と、再暗号シャッフル証明文622中のコミットメント614である変換情報保有コミットメント605　$g''[0,\Gamma];\Gamma=0,1$と、変換条件コミットメント609、912　$\psi[i],\phi[i],\omega,v'',v',v,u,u[0],u[i];i=1,\ldots,n$と、を挑戦値生成関数704に代入して挑戦値705を、

$c[0]=1$

$c[i]=Hash[i](g[\nu,\Gamma],g''[\nu,\Gamma],u,u[\nu],v,\phi[j],\psi[j],\omega,v',v'';\Gamma=0,1,2;\nu=0,\ldots,n;j=1,\ldots,n)i=1,\ldots,n$

と生成する。

【０１５８】変換情報保有検証処理710により、この挑戦値705を用いて、入力文列600と、変換情報保有コミットメント605と、出力暗号文列604である出力文列603と、応答618と、を用いて検証式、

$$\Pi_{\mu=0}^{n} g[\mu,\Gamma]^{r[\mu]}=\Pi_{\mu=0}^{n} g''[\mu,\Gamma]^{c[\mu]} /F_p \quad \Gamma=0,1$$

が成り立つことを確認706する。

【０１５９】変換条件検証処理711により、挑戦値705と、応答618と、変換条件コミットメント609、612と、を用いて検証式、

$$v''^{r'} v'^{r[0]} v^{\{\Sigma_{i=1}^{n} r[i]r[i]r[i]\}}=\omega v^{\{\Sigma_{i=1}^{n}(c[i]c[i]c[i]+\psi[i]c[i]c[i]+\phi[i]c[i])\}} /F_p$$

と検証式707

$$u^{r'}=\Pi_{i=1}^{n} u[i]^{r[i]r[i]} /F_p$$

が成り立つこと確認708する。

【０１６０】以上全ての検証式が成り立てば証明文を受理709する。

【０１６１】上記証明付再暗号シャッフル方法は、入力文列に対する再暗号シャッフル行列変換が少なくとも置換行列に属するシャッフル行列を持つ再暗号シャッフル行列により行われたことを保証する効果がある。この時、出力暗号文列$g''[i\Gamma];i=1,\ldots,n;\Gamma=0,1$は、

$g''[j,0]=g[i,0]^{w[i]}\ g[0,0]^{A[0,j]} /F_p$

$g''[j,1]=g[i,1]^{w[i]}\ g[0,1]^{A[0,j]} /F_p$

を出力した可能性を排除できない。ここで、$w[i]$が全て1の時が、再暗号シャッフルである。なお、$w[i];i=1,\ldots,n$は、F_q上の1の三乗根のいずれかをとる。

【０１６２】そこで、復号文として、F_q上の1の三乗根乗の自由度を許すか、平文に決められた記号を記して三乗根乗の自由度を消せば、本実施例をもって、証明付再暗号シャッフルを構成できる。

【０１６３】なお証明付再暗号シャッフル装置の変換情報保有コミットメント処理623、変換条件コミットメント生成処理625、応答生成処理624は、コンピュータ上で実行されるプログラムによりその処理・機能が実現される。また証明付再暗号シャッフル検証装置の変換情報保有検証処理710、変換条件検証処理711は、コンピュータ上で実行されるプログラムによりその処理・機能が実現される。この場合、該プログラムを記録した記録媒体（例えばＣＤ－ＲＯＭ、ＤＶＤ（digital versatile disk）、フロッピディスク媒体、ハードディスク媒体、磁気テープ媒体、半導体メモリ等のいずれか）から、該プログラムをコンピュータの主記憶にロードして実行することで、本発明を実施することができる。

【０１６４】［実施例(3)］本発明の実施例(3)として証明付再暗号シャッフル方法およびその検証方法について、図８、図９を参照して以下に説明する。以下で、$\Gamma=0,1$を取るものとする。また公開鍵は、$\eta[-1,\Gamma],\eta[0,\Gamma];\Gamma=0,1$の二組あり、どちらも同じ秘密鍵を持つとする。

【０１６５】再暗号シャッフル情報801として、再暗号シャッフル行列802、元係数803,805、係数基底804,806、準元係数807を以下のように準備する。

【０１６６】本実施例で用いる再暗号シャッフル行列802は、実施例(1)と実施例(2)のものとは大きさが異り、n+2行n+1列の行列である。

【０１６７】この再暗号シャッフル行列802を構成するシャッフル行列は、$A[i,j];i,j=1,\ldots,n$であり、再暗号秘密乱数は、$A[-1j],A[0,j];j=1,\ldots,n$の2×n成分であり、知識の隠蔽因子は、$A[\mu,0];\mu=-1,\ldots,n$のn+2成分である。これらの成分を、実施例(1)と同様に生成する。

【０１６８】元係数803　$r'[-1],r'[0]$、元係数805　ρ,ρ',ρ''、係数基底804　v、係数基底806　u、準元係数807　$\lambda[\mu];\mu=0,\ldots,n$に関しても、実施例(1)と同様な手法で、$r'[-1],r'[0],\rho,\rho',\rho'',\lambda[\mu];\mu=0,\ldots,n$には、1,0でない$F_q$上の数を、係数基底u,vには、位数qの$F_p$の元を生成する。

【０１６９】$r'[-1]\in_R Z_q,\neq 0,1$

$r'[0],\in_R Z_q,\neq 0,1$

$\rho\in_R Z_q,\neq 0,1$

$\rho'\in_R Z_q,\neq 0,1$

$\rho''\in_R Z_q,\neq 0,1$

$v\in_R Z_q,\neq 0,1,\text{s.t.}\quad v^q=1 /F_p$

$\lambda[\mu]\in_R Z_q,\neq 0,1\ \mu=0,\ldots,n$

$u\in_R Z_q,\neq 0,1,\text{s.t.}\quad u^q=1 /F_p$

【０１７０】入力暗号文列　$\eta[i,0],\eta[i,1];i=1,\ldots,n$と、公開鍵$\eta[-1,\Gamma],\eta[0,\Gamma];\Gamma=0,1$より、入力文列800　$g[\mu\Gamma];\mu=-1,\ldots,n;\Gamma=0,1$を、

$g[-1,\Gamma]=\eta[-1,\Gamma]\quad\Gamma=0,1$

$g[0,\Gamma]=\eta[0,\Gamma]\quad\Gamma=0,1$

$g[i,\Gamma]=\eta[i,\Gamma] /F_p\quad i=1,\ldots,n,\Gamma=0,1$

とする。

【０１７１】以下証明付再暗号シャッフル方法を用いる。

【０１７２】変換情報保有コミットメント生成処理832における再暗号シャッフル行列作用808により、上記再暗号シャッフル行列802を入力文列800に以下の様に作用させて、出力文列809$g''[\mu,\Gamma];\mu=0,\ldots,n;\Gamma=0,1$を、

$$g''[\mu,\Gamma]=\Pi_{\nu=1}^{n} g[\nu,\Gamma]^{A[\nu,\mu]} /F_p \quad \mu=0,\ldots,n,\ \Gamma=0,1$$

と生成する。ここで、$g''[i,\Gamma];i=1,\ldots,n;\Gamma=0,1$を、出力暗号文列810、$g''[0,\Gamma];\Gamma=0,1$を、変換情報保有コミットメント811とする。

【０１７３】変換条件コミットメント生成処理833、834における恒等式係数計算812、816により、元係数803、805　$r'[-1]$、$r'[0]$, ρ, ρ', ρ'', と、再暗号シャッフル行列802を用いて、恒等式係数817　$\psi[i]$, $\phi[i]$, $\phi[0]$, ρ, ρ', ρ''; $i=1,\ldots,n$　と、恒等式係数813　$\underline{\phi}[\nu]$, $r'[0]$, $r'[-1]$; $\nu=0,\ldots,n$　を計算818、812する。

【０１７４】$\rho=\rho$

$\rho'=\rho'$

$\rho''=\rho''$

$$\psi[i]=\Sigma_{j=1}^{n}(3A[j,0]+\rho''\lambda[j])A[j,i] /F_q \quad i=1,\ldots,n$$

$$\phi[i]=\Sigma_{j=1}^{n}(3A[j,0]A[j,0]A[j,i]+2\rho''\lambda[j]A[j,0]A[j,i])+\rho'A[0,i]+\rho A[-1,i] /F_q \quad i=1,\ldots,n$$

$$\phi[0]=\Sigma_{j=1}^{n}(A[j,0]A[j,0]A[j,0]+\rho''\lambda[j]A[j,0]A[j,0])+\rho''\lambda[0]+\rho'A[0,0]+\rho A[-1,0] /F_q$$

$r'[-1]=r'[-1]$

$r'[0]=r'[0]$

$$\underline{\phi}[0]=\Sigma_{j=1}^{n}A[j,0]A[j,0]+r'[0]A[0,0]+r'[-1]A[-1,0] /F_q$$

$$\underline{\phi}[i]=2\Sigma_{j=1}^{n}A[j,0]A[j,i]+r'[0]A[0,i]+r'[-1]A[-1,i] /F_q \quad i=1,\ldots,n$$

【０１７５】さらに係数基底804　vを用いて、隠蔽処理814、818により、恒等式係数813, 817　$r'[-1]$, $r'[0]$, $\underline{\phi}[0]$, $\phi[0]$, ρ, ρ', ρ''を、

$\omega=v^{\phi[0]} /F_p$

$v''=v^{\rho\cdot\cdot} /F_p$

$v'=v^{\rho\cdot} /F_p$

$\omega'=v^{\rho} /F_p$

とコミット819し、

$\underline{v}=v^{r'[-1]} /F_p$

$\underline{v'}=v^{r'[0]} /F_p$

$\underline{\omega}=v^{\phi[0]} /F_p$

とコミット815する。

【０１７６】さらに、係数基底806 uを用いて、準元係数807　$\lambda[\mu];\mu=0,\ldots,n$を、

$u[0]=u^{\lambda[0]} /F_q$

$u[i]=u^{\lambda[i]} /F_q \quad i=1,\ldots,n$

とコミット821、820する。

【０１７７】以上より、$\underline{\phi}[i]$, $\underline{v'}$, $\underline{v}$, $\underline{\omega}$, $\psi[i]$, $\phi[i]$, ω, v'', v', ω', v, u, $u[0]$, $u[i];i=1,\ldots,n$を変換条件コミットメント822とする。

【０１７８】ここで、コミットメント823を，変換情報保有コミットメント811と変換条件コミットメント822とする。

【０１７９】応答生成処理835により、以上の入力文列800と、出力暗号文列810と、コミットメント823を、挑戦値生成関数824の引数として、挑戦値825を、

$c[0]=1$

$c[i]=Hash[i](g[\mu,\Gamma],g''[\nu,\Gamma],u[\nu],u,\phi[j],\psi[j],\omega,\omega',v',v'',v,\underline{\phi}[j],\underline{\omega},\underline{v'},\underline{v};\mu=-1,\ldots,n;\nu=0,\ldots,n;j=1,\ldots,n;\Gamma=0,1,2)i=1,\ldots,n$

と生成し、この挑戦値825から、再暗号シャッフル行列802を用いて、応答827を、

$$r[\mu]=\Sigma_{\nu=0}^{n}A[\mu,\nu]^{c[\nu]} /F_q \quad \mu=-1,\ldots,n$$

と生成826する。

【０１８０】さらに、準応答829を、準元係数807 $\lambda[\mu];\mu=0,\ldots,n$と応答827より、

$$r'=\lambda[0]+\Sigma_{i=1}^{n}\lambda[i]r[i]r[i] /F_q$$

と生成828する。

【０１８１】以上のコミットメント823と、応答827と、準応答829とを再暗号シャッフル証明文831として出力し、再暗号シャッフルの結果として、出力暗号文列810を出力する。

【０１８２】検証方法について図９を参照して説明する。

【０１８３】再暗号シャッフル検証方法により、入力文列800　$g[\mu,\Gamma];\mu=-1,\ldots,n;\Gamma=0,1$と、出力暗号文列810　$g''[i,\Gamma];i=1,\ldots,n;\Gamma=0,1$と、再暗号シャッフル証明文831中コミットメント823の変換情報保有コミットメント811　$g''[0,\Gamma];\Gamma=0,1$と、変換条件コミットメント815、819、821　$\underline{\phi}[i]$, $\underline{v'}$, $\underline{v}$, $\underline{\omega}$, $\psi[i]$, $\phi[i]$, ω, v'', v', ω', v, u, $u[0]$, $u[i];i=1,\ldots,n$と、を、挑戦値生成関数900に代入して、挑戦値901を、

$c[0]=1$

$c[i]=Hash[i](g[\mu,\Gamma],g''[\nu,\Gamma],u[\nu],u,\phi[j],\psi[j],\omega,\omega',v',v'',v,\underline{\phi}[j],\underline{\omega},\underline{v'},\underline{v};\mu=-1,\ldots,n;\nu=0,\ldots,n;j=1,\ldots,n;\Gamma=0,1,2)i=1,\ldots,n$

と生成する。

【０１８４】変換情報保有検証処理907により、この挑戦値901を用いて入力文列800と、変換情報保有コミットメント811と、出力暗号文列810である出力文列809と、応答827とを用いて検証式、

$$\Pi_{\mu=1}^{n} g[\mu,\Gamma]^{r[\mu]}=\Pi_{\mu=0}^{n} g''[\mu,\Gamma]^{c[\mu]} /F_p \quad \Gamma=0,1$$

が成り立つことを確認902する。

【０１８５】変換条件検証処理908、909により、挑戦値901と応答827と準応答829と変換条件コミットメント815、819、821とを用いて、検証式

$v''^{r'} v'^{r[0]} \omega'^{r[-1]} v\hat{}\{\Sigma_{i=1}{}^{n} r[i]r[i]r[i]\}$
$=\omega v\hat{}\{\Sigma_{i=1}{}^{n}(c[i]c[i]c[i]+\psi[i]c[i]c[i]+\phi[i]c[i])\}$ /F_p
が成り立つことを確認904し、検証式
$u^{r'}=u[0]\Pi_{i=1}{}^{n} u[i]^{r[i]r[i]}$ /F_p
が成り立つことを確認905し、検証式
$\underline{v}'^{r[0]}\ \underline{v}^{r[-1]}\ v\hat{}\{\Sigma_{i=1}{}^{n} r[i]r[i]\}=\underline{\omega}v\hat{}\{\Sigma_{i=1}{}^{n}(c[i]c[i]+\underline{\phi}[i]c[i])\}$ /F_p
が成り立つこと確認903する。

【０１８６】以上全ての検証式が成り立てば証明文を受理906する。

【０１８７】上記証明付再暗号シャッフル方法は、入力文列に対する再暗号シャッフル行列変換が少なくとも置換行列に属するシャッフル行列を持つ再暗号シャッフル行列により行われたことを保証する効果がある。これは再暗号シャッフルが行われたことを意味し、本実施例は証明付再暗号シャッフルである。

【０１８８】なお証明付再暗号シャッフル装置の変換情報保有コミットメント処理832、変換条件コミットメント生成処理833、834、応答生成処理835は、コンピュータ上で実行されるプログラムによりその処理・機能が実現される。また証明付再暗号シャッフル検証装置の変換情報保有検証処理907、変換条件検証処理908、909は、コンピュータ上で実行されるプログラムによりその処理・機能が実現される。この場合、該プログラムを記録した記録媒体（例えばＣＤ－ＲＯＭ、ＤＶＤ（digital versatile disk）、フロッピディスク媒体、ハードディスク媒体、磁気テープ媒体、半導体メモリ等のいずれか）から、該プログラムをコンピュータの主記憶にロードして実行することで、本発明を実施することができる。

【０１８９】［実施例(4)］本発明の実施例(4)の証明付再暗号シャッフル方法およびその検証方法について図１０、図１１を参照して以下に説明する。以下で、$\Gamma=0,1$を取るものとする。また公開鍵は、$\eta[0,\Gamma];\Gamma=0,1$の一組である。

【０１９０】再暗号シャッフル情報1006として、再暗号シャッフル行列1001と、第２の情報隠蔽因子1004と、元係数1002、1005と、係数基底1003、1008と、準元係数1007を以下のように準備する。

【０１９１】まず再暗号シャッフル行列1001に関しては、前記実施例(1)と同様に生成し、これを$\underline{A}[\mu,\nu];\mu,\nu=0,\ldots,n$とする。

【０１９２】さらに、第２の情報隠蔽因子1004　$A[\nu,0];\nu=0,\ldots,n$を同様に生成する。

【０１９３】元係数1005　ρ',ρ''、元係数1002　$r'[0]$、係数基底1003　v、係数基底1008u、準元係数1007　$\lambda[\mu];\mu=0,\ldots,n$　に関しても、実施例(1)と同様な手法で、$r'[0],\rho',\rho'',\lambda[\mu];\mu=0,\ldots,n$には1,0でない$F_q$上の数を、係数基底u,vには位数qの$F_p$の元を生成する。

【０１９４】$\rho'\in_R Z_q,\neq 0,1$
$\rho''\in_R Z_q,\neq 0,1$
$r'[0],\in_R Z_q,\neq 0,1$
$v\in_R Z_p,\neq 0,1,$ s.t. $v^q=1$ /F_p
$\lambda[\mu]\in_R Z_q,\neq 0,1\ \mu=0,\ldots,n$
$u\in_R Z_p,\neq 0,\ 1,$ s.t. $u^q=1$ /F_p

【０１９５】入力暗号文列　$\eta[i,0],\eta[i,1];i=1,\ldots,n$と、公開鍵　$\eta[0,\Gamma];\Gamma=0,1$より、入力文列1000　$g[\mu\Gamma];\mu=0,\ldots,n;\Gamma=0,1$を、
$g[0,\Gamma]=\eta[0,\Gamma]\quad\Gamma=0,1$
$g[i,\Gamma]=\eta[i,\Gamma]\quad i=1,\ldots,n,\ \Gamma=0,1$
とする。

【０１９６】以下、証明付再暗号シャッフル方法を用いる。

【０１９７】変換情報保有コミットメント生成処理1042における再暗号シャッフル行列作用1009により、上記暗号シャッフル行列1001を入力文列1000に以下の様に作用させて、出力文列1010　$g''[\mu,\Gamma];\mu=0,\ldots,n;\Gamma=0,1$を、
$g''[\mu,\Gamma]=\Pi_{\nu=1}{}^{n} g[\nu,\Gamma]^{A[\nu,\mu]}$ /F_p　　$\mu=0,\ldots,n,\ \Gamma=0,1$
と生成する。

【０１９８】ここで、$g''[i,\Gamma];i=1,\ldots,n;\Gamma=0,1$、出力暗号文列1011、$g''[0,\Gamma];\Gamma=0,1$を第1の変換情報保有コミットメント1012とする。

【０１９９】第２の変換情報保有コミットメント生成処理1044により、さらに入力文列1000から選択1018して、第２の入力文列1019を$g[\mu,\Gamma']$とする。ここでは、$\Gamma'=0$とする。

【０２００】第２の変換情報保有コミットメント1021　$\underline{g''}[0,\Gamma']$を、
$\underline{g''}[0,\Gamma']=\Pi_{\nu=1}{}^{n} g[\nu,\Gamma']^{A[\nu,0]}$ /F_p　$\Gamma=0$or1
と生成1020する。

【０２０１】変換条件コミットメント生成処理1045における恒等式係数計算1022により、元係数1005　ρ',ρ''と、再暗号シャッフル行列1001と、を用いて、恒等式係数1023　$\psi[i],\phi[i],\phi[0],\rho',\rho'';i=1,\ldots,n$を、
$\rho'=\rho'$
$\rho''=\rho''$
$\psi[i]=\Sigma_{j=1}{}^{n}(3A[j,0]+\rho''\lambda[j])A[j,i]$ /F_q　　$i=1,\ldots,n$
$\phi[i]=\Sigma_{j=1}{}^{n}(3A[j,0]A[j,0]A[j,i]+2\rho''\lambda[j]A[j,0]A[j,i])+\rho'A[0,i]$ /F_q $i=1,\ldots,n$
$\phi[0]=\Sigma_{j=1}{}^{n}(A[j,0]A[j,0]A[j,0]+\rho''\lambda[j]A[j,0]A[j,0])+\rho''\lambda[0]+\rho'A[0,0]$ /F_q
と生成する。

【０２０２】さらに、係数基底1003vを用いて、隠蔽処理1024により、恒等式係数1023$\phi[0],\rho',\rho''$を、
$\omega=v^{\phi[0]}$ /F_p

$v''=v^{\rho}$ /F_p
$v'=v^{\rho}$ /F_p
とコミット1025する。
【０２０３】さらに、係数基底1008uを用いて、準元係数1007$\lambda[\mu]$;$\mu=0,\ldots,n$を、
$u[0]=u^{\lambda[0]}$ /F_q
$u[i]=u^{\lambda[i]}$ /F_q　　$i=1,\ldots,n$
とコミット1027する。
【０２０４】変換条件コミットメント生成処理1043における恒等式係数計算1013により、元係数1002r'[0]と再暗号シャッフル行列1001と第２の情報隠蔽因子1004とを用いて恒等式係数1014　$\underline{\phi}[\nu]$,r'[0];$\nu=0,\ldots,n$を、
r'[0]=r'[0]
$\underline{\phi}[0]=\Sigma_{j=1}{}^{n}\underline{A}[j,0]\underline{A}[j,0]+r'[0]\underline{A}[0,0]$ /F_q
$\underline{\phi}[i]=2\Sigma_{j=1}{}^{n}\underline{A}[j,0]A[j,i]+r'[0]A[0,i]$ /F_q　$i=1,\ldots,n$
と生成する。
【０２０５】さらに係数基底1003　vを用いて、隠蔽処理1015により、恒等式係数1014　r'[0],$\underline{\phi}[0]$を、
$\underline{v'}=v^{r'[0]}$ /F_p
$\underline{\omega}=v^{\underline{\phi}[0]}$ /F_p
とコミット1016する。
【０２０６】以上により、第1の変換条件コミットメント1028を、$\psi[i]$,$\phi[i]$,ω,v'',v',v,u,u[0],u[i];i=1,...,nとする。第２の変換条件コミットメント1016を、$\underline{\phi}[i]$,$\underline{v'}$,$\underline{\omega}$,v;i=1,...,nとする。
【０２０７】ここで、第１のコミットメント1017を、第１の変換情報保有コミットメント1012と第１の変換条件コミットメント1028とし、第２のコミットメント1029を第２の変換情報保有コミットメント1021と第２の変換条件コミットメント1016とする。
【０２０８】応答生成処理1046により、以上の入力文列1000と、出力暗号文列1011と、第1コミットメント1017とを、挑戦値生成関数1030の引数として、第１の挑戦値1031を、
c[0]=1
$c[i]=Hash[i](g[\nu,\Gamma],g''[\nu,\Gamma],u[\nu],u,\phi[j],\psi[j],\omega,v',v'',v;\nu=0,\ldots,n;j=1,\ldots,n;\Gamma=0,1)$　　$i=1,\ldots,n$
と生成し、この挑戦値1031から、再暗号シャッフル行列1001を用いて、第１の応答1033を、
$r[\mu]=\Sigma_{\nu=0}{}^{n}A[\mu,\nu]c[\nu]$ /F_q　　$\mu=0,\ldots,n$
と生成1032する。
【０２０９】さらに、準応答1039を、準元係数1007$\lambda[\mu]$;$\mu=0,\ldots,n$と、応答1033より、
$r'=\lambda[0]+\Sigma_{i=1}{}^{n}\lambda[i]r[i]r[i]$ /F_q
と生成1038する。
【０２１０】応答生成処理1047により、第２の入力文列1019と、出力暗号文列1011と、第２コミットメント1029とを、挑戦値生成関数1034の引数として、第２の挑戦値1035を、
c[0]=1
$\underline{c}[i]=Hash[i](g[\nu,\Gamma'],\underline{g''}[0,\Gamma'],g''[j,\Gamma'],\underline{\phi}[j],\underline{\omega},\underline{v'};\nu=0,\ldots,n;j=1,\ldots,n;\Gamma'=0)$　$i=1,\ldots,n$
と生成し、この挑戦値1035から、再暗号シャッフル行列1001と、第２の情報隠蔽因子1004を用いて、第２の応答1037を、
$\underline{r}[\mu]=\underline{A}[\mu,0]+\Sigma_{i=1}{}^{n}A[\mu,i]\underline{c}[i]$ /F_q　　$\mu=0,\ldots,n$
と生成1036する。
【０２１１】以上のコミットメント1017、1029と、応答1033、1037と、準応答1039と、を、再暗号シャッフル証明文1040として出力し、再暗号シャッフルの結果として出力暗号文列1011を出力する。
【０２１２】検証方法について、図１１を参照して説明する。
【０２１３】再暗号シャッフル検証方法により、入力文列1000と、出力暗号文列1011と、再暗号シャッフル証明文1040の第1のコミットメント1012、1025、1027を、挑戦値生成関数1100に代入して、第1の挑戦値1101を、
c[0]=1
c[i]=Hash[i](入力文列、出力暗号文列、第1のコミットメント)　i=1,...,nと生成する。
【０２１４】さらに、第２の入力文列1019と、再暗号シャッフル証明文1040の第２のコミットメント1016、1021と、出力暗号文列1011とを、挑戦値生成関数1108に代入して第２の挑戦値1109を
c[0]=1
c[i]=Hash[i](第２の入力文列、出力暗号文列、第２のコミットメント)　i=1,...,nと生成する。
【０２１５】変換情報保有検証処理1112により、第１の挑戦値1101を用いて、入力文列1000と、第１の変換情報保有コミットメント1012と、出力暗号文列1011と、第１の応答1033と、を用いて検証式、
$\Pi_{\mu=0}{}^{n}g[\mu,\Gamma]^{r[\mu]}=\Pi_{\mu=0}{}^{n}g''[\mu,\Gamma]^{c[\mu]}$ /F_p　$\Gamma=0,1$
が成り立つことを確認1103する。
【０２１６】変換情報保有検証処理1113により、第２の挑戦値1109を用いて第２の入力文列1019と、第２の変換情報保有コミットメント1021と、出力暗号文列1011と、第２の応答1037と、を用いて第２の知識検証式、
$\Pi_{\mu=0}{}^{n}g[\mu,\Gamma']^{r[\mu]}\underline{}^{c[i]}=\underline{g''}[0,\Gamma']\Pi_{i=1}{}^{n}g''[i,\Gamma']$ /F_p　$\Gamma'=0$
が成り立つことを確認1105する。
【０２１７】変換条件検証処理1111により、第1の挑戦値1101と、第1の応答1033と、第1の変換条件コミットメント1025とを用いて、検証式1102、
$v''^{r'}v'^{r'[0]}\omega'^{r'[-1]}v^{\hat{}}\{\Sigma_{i=1}{}^{n}r[i]r[i]r[i]\}=\omega v^{\hat{}}\{\Sigma_{i=1}{}^{n}(c[i]c[i]c[i]+\psi[i]c[i]c[i]+\phi[i]c[i])\}$ /F_p
と、準応答1039と、準応答コミットメント1027と、第1

の応答1033と、検証式1107、
$u'^{r'} = u[0]\prod_{i=1}{}^{n} u[i]^{r[i]r[i]} \quad /F_p$
が成り立つことを確認する。

【０２１８】変換条件検証処理1114により、第２の挑戦値1109と、第２の応答1037と、第２の変換条件コミットメント1016とを用いて、検証式1106、
$\underline{v}'^{r[0]}\ \hat{v}\{\Sigma_{i=1}{}^{n}\underline{r}[i]\underline{r}[i]\} = \underline{\omega}\hat{v}\{\Sigma_{i=1}{}^{n}(\underline{c}[i]\underline{c}[i] + \underline{\phi}[i]\underline{c}[i])\}\ /F_p$
が成り立つことを確認する。

【０２１９】以上全ての検証式が成り立てば証明文を受理1110する。

【０２２０】上記証明付再暗号シャッフル方法は、入力文列に対する再暗号シャッフル行列変換が少なくとも置換行列に属するシャッフル行列を持つ再暗号シャッフル行列により行われたことを保証する効果がある。これは再暗号シャッフルが行われたことを意味し、本実施例は証明付再暗号シャッフルである。

【０２２１】なお証明付再暗号シャッフル装置の変換情報保有コミットメント処理1042、変換条件コミットメント生成処理1043、1045、応答生成処理1046、1047は、コンピュータ上で実行されるプログラムによりその処理・機能が実現される。また証明付再暗号シャッフル検証装置の変換情報保有検証処理1112、1113、変換条件検証処理1111、1114は、コンピュータ上で実行されるプログラムによりその処理・機能が実現される。この場合、該プログラムを記録した記録媒体（例えばＣＤ－ＲＯＭ、ＤＶＤ（digital versatile disk）、フロッピディスク媒体、ハードディスク媒体、磁気テープ媒体、半導体メモリ等のいずれか）から、該プログラムをコンピュータの主記憶にロードして実行することで、本発明を実施することができる。

【０２２２】［実施例(5)］本発明の実施例(5)をなす入力文列生成方法について、図１２を参照して、以下に説明する。Γは、0、1、2の値をとる。

【０２２３】公開鍵302　g[0,0]、g[0,1]に対応する秘密鍵xは、t人の証明者により分散所持されている。

【０２２４】この時、秘密鍵x[Λ]；Λ=1,...,tにより、各証明者の公開鍵を、g[0,0]、$g[0,1,\Lambda]=g[0,0]^{x[\Lambda]}$；Λ=1,...,tとし、全体の公開鍵は、g[0,0]、$g[0,1]=\prod_{k=1}{}^{t} g[0,1,\Lambda]$とする。

【０２２５】入力暗号文列301　η[i,0]、η[i,1]；i=1,...,nと、公開鍵302　η[0,0]、η[0,1]が入力され、公開鍵302およびElgamal領域変数p、qより、基底生成関数1200で、入力ベクトル1201を生成し、入力文列300　g[μ,Γ]；μ=0,...,n；Γ=0,1,2を、
g[0,Γ]=η[0,Γ]　　Γ=0,1
g[i,Γ]=η[i,Γ]/F_p　i=1,...,n,　Γ=0,1
g[μ,2]=Hash'[μ](p,q,η[0,0],g[0,1,Λ]；Λ=1,...,t) μ=0,...,n
とする。

【０２２６】前記実施例(1)乃至実施例(4)に、本実施例の入力文列生成方法を適用する場合、これらの実施例におけるΓの値をとる範囲を、全て0、1から、0、1、2に変更する。この新たに導入された、入力暗号文でも公開鍵でもないΓ=２の成分が、入力暗号文生成者にも意図できない入力文列の成分となり、証明者が生成できる応答に制限を課す働きをし、入力暗号文生成者と再暗号シャッフル証明文生成者とが共謀して再暗号シャッフル証明文の偽造を行うことを阻止する。

【０２２７】また前記実施例(3)に、本実施例の入力文列生成方法を適用する場合、入力文列を、g[-1,Γ]まで拡張して、公開鍵　g[-1,0]、g[-11]、η[0,0]、η[0,1]より、
g[-1,Γ]=η[-1,Γ]　　Γ=0,1
g[0,Γ]=η[0,Γ]　　Γ=0,1
g[i,Γ]=η[i,Γ]/F_p　　i=1,...,n, Γ=0,1
g[μ,2]=Hash'[μ](p,q,η[0,0],g[0,1,Λ]；Λ=1,...,t)　μ=-1,...,n
とする。

【０２２８】また実施例(4)に、本実施例の入力文列生成方法を適用する場合、Γ'=２とし、第２の情報隠蔽因子より、第２の変換情報保有コミットメントを、
$\underline{g}''[02]=\prod_{\nu=1}{}^{n} g[\nu,2]^{\Lambda'[\nu,0]} \quad /F_p$
$\underline{g}''[i2]=\prod_{\nu=1}{}^{n} g[\nu,2]^{\Lambda[\nu,i]} \quad /F_p \quad i=1,...,n$
と変更する。

【０２２９】さらに、証明付再暗号シャッフル方法または再暗号シャッフル検証方法において、第２の入力文列　g[μ,Γ']；　Γ=２と、第２のコミットメントとを挑戦値生成関数の引数として、第２の挑戦値を、
c[0]=1
c[i]=Hash[i](g[ν,2],g''[ν,2],φ[j],ω,v'；；ν=0,...,n；j=1,...,n；Γ=0,1)　i=1,...,n
と変更する。

【０２３０】さらに、変換情報保有検証処理における第２の知識検証式は、
$\prod_{\mu=0}{}^{n} g[\mu,2]^{r[\mu]} = \prod_{\mu=0}{}^{n} g''[\mu,\Gamma']^{c[\mu]} \quad /F_p$
と変更する。

【０２３１】［実施例(6)］本発明の実施例(6)をなす入力文列生成方法について、図１３と図１４を参照して説明する。Γは、0、1の値をとる。

【０２３２】前記実施例(5)と同様に、秘密鍵xは、t人の証明者により分散所持されている。各証明者Λ；Λ=1,...,tは、証明付公開鍵列方法1304により、入力暗号文列301 η[i,0]、η[i,1]；i=1,...,nと、公開鍵302　η[0,0]、η[0,1]とを、共通初期値1310として、秘密鍵1301x[Λ]と、疑似秘密鍵1302α[Λ]と、を公開鍵列情報1300として入力し、分散公開鍵列対1305　g'[μ,1,Λ]；μ=0,...,nと、公開鍵列証明文1306と、を得る。

【０２３３】公開鍵列検証方法1307により、各証明者の出力した分散公開鍵列対1305と、公開鍵列証明文と、共

通初期値1310とから、分散公開鍵列1305の正当性が検証されたら、前処理方法により、各証明者の分散公開鍵列対1305　g'[μ,1,Λ];μ=0,...,n;Λ=1,...,tを合わせて、公開鍵列対140　3g'[μ,1,Λ];μ=0,...,nを、

$g'[\mu,1]=\Pi_{\Lambda=1}{}^{t}g'[\mu,1,\Lambda]\ /F_p \qquad \mu=0,\ldots,n$

とする。ここで、g'[0,1]=η[0,1]に入れ替える。

【０２３４】共通初期値である入力暗号文列301　η[i,0]、η[i,1];i=1,...,nと、公開鍵302η[0,0]、η[0,1]とから、公開鍵列底1401　g'[μ0];μ=0,...,nを、

g'[0,0]=η[0,0]

g'[i,0]=Hash'[i](η[0,0],η[0,1,Λ],η[j,Γ];Λ=1,...,t;Γ=0,1;j=1,...,n;)　i=1,...,n

と生成1400する。ここでも、公開鍵列対1403と同様にg'[0,0]が入れ替えている。

【０２３５】公開鍵列底1401と公開鍵列対1403を合わせて、公開鍵列1404　g'[μ,Γ];μ=0,...,n,Γ=0,1とする。

【０２３６】公開鍵列1404と、入力暗号文列301と、公開鍵302とから、入力文列300　g[μ,Γ];μ=0,...,n;Γ=0,1を、

g[0,Γ]=η[0,Γ]　　Γ=0,1

$g[i,\Gamma]=\eta[i,\Gamma]g'[i,\Gamma]\ /F_p \qquad i=1,\ldots,n,\ \Gamma=0,1$

とする（前処理1402）。

【０２３７】前記実施例(3)に、本実施例の入力文列生成方法を適用する場合、入力暗号文列　η[i,Γ];i=1,...,n;Γ=0,1と、公開鍵　η[0,Γ];Γ=0,1に対して、公開鍵列　g'[μ,Γ];μ=-1,...,n;Γ=0,1を生成する。ただし、g'[0,Γ];Γ=0,1は、公開鍵に等しい。そして、入力文列　g[μ,Γ];μ=-1,...,n;Γ=0,1を、

g[-1,Γ]=η[0,Γ]　　Γ=0,1

$g[i,\Gamma]=\eta[i,\Gamma]g'[i,\Gamma]\ /F_p \qquad i=0,\ldots,n,\ \Gamma=0,1$

とする。

【０２３８】本実施例では、新たに生成された公開鍵列が、入力暗号文生成者にも意図できないため、入力暗号文にそれを乗じた入力文列の成分も意図できない。そのため、証明者が生成できる応答に制限を課す働きをなし、入力暗号文生成者と再暗号シャッフル証明文生成者とが共謀して再暗号シャッフル証明文の偽造を行うことを阻止する。

【０２３９】なお図１３に示した、証明付公開鍵列装置1304、前処理装置1309、公開鍵列検証装置1307は、コンピュータ上で実行されるプログラムによりその処理・機能が実現される。この場合、該プログラムを記録した記録媒体（例えばＣＤ－ＲＯＭ、ＤＶＤ（digital versatile disk）、フロッピディスク媒体、ハードディスク媒体、磁気テープ媒体、半導体メモリ等のいずれか）から、該プログラムをコンピュータの主記憶にロードして実行することで、本発明を実施することができる。

【０２４０】［実施例(7)］本発明の実施例(7)の入力文列生成方法について、図１５乃至図１８を参照て説明する。Γは、0、1の値をとる。前記実施例(5)と同様に、秘密鍵1502　xは、t人の証明者により、分散所持されている。

【０２４１】各証明者Λ;Λ=1,...,tは、証明付公開鍵列方法1504により、Elgamal領域変数を共通初期値1500として、秘密鍵1502　x[Λ]と、疑似秘密鍵1503　α[Λ]と、を、公開鍵列情報1501として入力し、分散公開鍵列対1505　g'[μ,1,Λ];μ=0,...,nと、公開鍵列証明文1506と、を得る。

【０２４２】公開鍵列検証方法1507により、各証明者の出力した分散公開鍵列対1505と、公開鍵列証明文1506と、共通初期値1500より、分散公開鍵列対1505の正当性が検証1508されたら、各証明者の分散公開鍵列対150　5g'[μ,1,Λ];μ=0,...,n;Λ=1,...,tを合わせて、公開鍵列対1509　g'[μ,1,Λ];μ=0,...,nを、

$g'[\mu,1]=\Pi_{\Lambda=1}{}^{t}g'[\mu,1,\Lambda]\ /F_p \qquad \mu=0,\ldots,n$

とする。

【０２４３】共通初期値1500から公開鍵列底g'[μ,0];μ=0,...,nを、

$g'[\mu,0]=\mathrm{Hash}'[\mu](p,q) \qquad \mu=0,\ldots,n$

と生成する。

【０２４４】公開鍵列底と公開鍵列対1509を合わせて、公開鍵列1611　g'[μΓ];μ=0,...,n,Γ=0,1とする。

【０２４５】各入力暗号文生成者i=1....,nは、証明付暗号化方法1606により、平文1602m[i]と、個別公開鍵1601g'[i,Γ];Γ=0,1と、秘密乱数1604s[i]と、疑似秘密乱数1605s'[i]とより、入力暗号文1607η[i,Γ];Γ=0,1を、

$\eta[i,0]=g'[i,0]^{s[i]} \ /F_p$

$\eta[i,1]=m[i]g'[i,1]^{s[i]} \ /F_p$

と生成する。

【０２４６】またコミットメント（疑似暗号文底1704）、挑戦値1707、応答1709を順に以下のように生成し、

$\eta[i2]=g'[i,0]^{s'[i]} \ /F_p$

c'[i]=Hash[0](η[i,0],η[i,1],η[i2])

$\theta'[i]=c'[i]s[i]+s'[i] \ /F_q$

疑似暗号文底1704と応答1709とを暗号化証明文1608とする。

【０２４７】暗号化検証装置により、全ての入力暗号文1607と暗号化証明文1608に関して、

c'[i]=Hash[0](η[i,0],η[i,1],η[i2])

と、挑戦値1801を求め、これに応答1709を用いて、検証式1802

$\eta[i,0]^{\theta'[i]}=\eta[i,1]^{c'[i]}\eta[i2] \ /F_p$

が成り立つことを確認1610する。入力暗号文1607全ての正当性が確認されたら、入力暗号文323　η[i,Γ];Γ=0,1と、共有公開鍵1600　g'[0,Γ];Γ=0,1とより入力文列300を、

g[0,Γ]=g'[0,Γ]

$g[i,\Gamma]=\eta[i,\Gamma] \quad i=1,\ldots,n$

とする。

【０２４８】前記実施例(3)に、本実施例の入力文列生成方法を適用する場合は、

$g[-1\Gamma]=g'[-1,\Gamma]$

$g[0,\Gamma]=g'[0,\Gamma]$

$g[i,\Gamma]=\eta[i,\Gamma]\quad i=1,\ldots,n$

とする。

【０２４９】本実施例では、始めに生成された公開鍵列が、入力暗号文生成者にも意図できないため、これを基に暗号化したことが主命されている入力暗号文の成分も、意図できない。そのため、証明者が生成できる応答に制限を課す働きをし、入力暗号文生成者と再暗号シャッフル証明文生成者とが共謀して再暗号シャッフル証明文の偽造を行うことを阻止する。

【０２５０】なお図１５に示した、証明付公開鍵列装置1504、公開鍵列検証装置1507は、コンピュータ上で実行されるプログラムによりその処理・機能が実現される。また図１６乃至図１８に示した、証明付暗号化装置1606、暗号化検証装置1609は、コンピュータ上で実行されるプログラムによりその処理・機能が実現される。この場合、該プログラムを記録した記録媒体（例えばＣＤ－ＲＯＭ、ＤＶＤ（digital versatile disk）、フロッピディスク媒体、ハードディスク媒体、磁気テープ媒体、半導体メモリ等のいずれか）から、該プログラムをコンピュータの主記憶にロードして実行することで、本発明を実施することができる。

【０２５１】［実施例(8)］本発明の実施例(8)における証明付公開鍵列方法について図１９と図２０とを参照して説明する。

【０２５２】共通初期値eと、秘密鍵1902　xと、疑似秘密鍵1903　αとが、公開鍵列情報1901として、入力される。

【０２５３】共通初期値1900から、公開鍵列底1905　$g'[\mu 0];\mu=0,\ldots,n$を、

$g'[\mu,0]=Hash'[\mu](e)\quad \mu=0,\ldots,n$

と生成1904する。

【０２５４】これから、秘密鍵1902　xと、疑似秘密鍵1903αとにより、(分散)公開鍵列対1907　$g'[\mu,1];\mu=0,\ldots,n$が、

$g'[\mu,1]=g'[\mu,0]^{x}\ /F_p\qquad \mu=0,\ldots,n$

と生成1906され、疑似公開鍵列対1909が、

$g'[\mu,2]=g'[\mu,0]^{\alpha}\ /F_p\qquad \mu=0,\ldots,n$

と生成1908される。

【０２５５】挑戦値1912、応答1914を順に、

$c''=Hash[0](g'[\mu,0],g'[\mu,2]\ ;\mu=0,\ldots,n)$

$\theta=c''x+\alpha\ /F_q$

と生成し、疑似公開鍵列1909と応答1914を公開鍵列証明文1915とする。

【０２５６】公開鍵列検証方法により、挑戦値2003を、

$c''=Hash[0](g'[\mu,0],g'[\mu,2];\mu=0,\ldots,n)$

と生成2000し、応答1914を用いて検証式、

$g'[\mu,0]^{\theta}=g'[\mu,0]^{c''}\ g'[\mu,2]\ /F_p\qquad \mu=0,\ldots,n$

を検証2004する。

【０２５７】本実施例では、始めに生成された公開鍵列底、誰にも意図できないため、これを元に作られた公開鍵列の成分も意図できない。

【０２５８】なお図１９、図２０に示した、証明付公開鍵列装置、公開鍵列検証装置は、コンピュータ上で実行されるプログラムによりその処理・機能が実現される。この場合、該プログラムを記録した記録媒体（例えばＣＤ－ＲＯＭ、ＤＶＤ（digital versatile disk）、フロッピディスク媒体、ハードディスク媒体、磁気テープ媒体、半導体メモリ等のいずれか）から、該プログラムをコンピュータの主記憶にロードして実行することで、本発明を実施することができる。

【０２５９】［実施例(9)］本発明の実施例(9)として、証明付復号について説明する。前記実施例(5)と同様に、秘密鍵　xは、t人の証明者により分散所持されている。

【０２６０】$\Lambda;\Lambda=1,\ldots,t$番目の証明者は、$\Lambda-1$番目の証明者による部分復号の結果を入力して、それを部分復号する。Λ番目の証明者による部分復号の結果が復号文である。ただし、0番目の証明者による部分復号の結果は、上記最終的な再暗号シャッフルの出力のことである。

【０２６１】ここで、Λ番目の証明者の行う証明付部分復号(部分復号およびその証明文の提出)について説明する。

【０２６２】疑似乱数発生器により1,0でないF_q上の数を$\beta[\Lambda]$を作成する。

【０２６３】$\beta[\Lambda]\in_R Z_q,\neq 0,1$

【０２６４】また、自身の公開鍵　$g[0,0]$、$g'[0,1,\Lambda]$を、$g[0,0]$、$g[0,1]$と、入力された暗号文列を、$g[i,\Gamma];i=1,\ldots,n,\ \Gamma=0,1$とし、自身の公開鍵と秘密鍵$x[\Lambda]$から、部分復号基底$G[\mu,0\Lambda];\mu=0,\ldots,n$と、疑似部分復号基底$G[\mu,1,\Lambda];\mu=0,\ldots,n$を、

$G[\mu,0,\Lambda]=g[\mu,0]^{x[\Lambda]}\ /F_p\qquad \mu=0,\ldots,n$

$G[\mu,1,\Lambda]=g[\mu,0]^{\beta[\Lambda]}\ /F_p\qquad \mu=0,\ldots,n$

と生成する。コミットメントとして、$g[\mu,\Gamma\Lambda];\mu=0,\ldots,n,\Gamma=0,1,\Lambda=0,\ldots,t$を出力する。

【０２６５】$g[0,1,\Lambda]=g[0,0]^{x[\Lambda]}=G[0,0,\Lambda]$は、公開鍵と重複しているが同じものが計算されている。

【０２６６】挑戦値を、

$c[\Lambda]=Hash[0](g[\mu,0],G[\mu,\Gamma,\Lambda];\mu=0,\ldots,n;\Gamma=0,1)$

と生成し、これを用いて、応答$r[\Lambda]$を

$r[\Lambda]=\beta[\Lambda]+c[\Lambda]x[\Lambda]\ /F_q$

と生成して出力する。部分復号基底、疑似部分復号基底と応答を証明付部分復号の証明文として出力する。

【０２６７】部分復号を

g[i,0]→g[i,0]　i=1,...,n
g[i,1]→g[i,1]/G[i,0,Λ]　/Fp　i=1,...,n
として出力する。

【０２６８】検証処理は、入力暗号文列と証明文中とより挑戦値を、
c[Λ]=Hash[0](g[μ,0],G[μ,Γ,Λ];μ=0,...,n;Γ=0,1)
と生成し、証明文中の応答、入力暗号文列、分復号基底、疑似部分復号基底を用いて、
$g[\mu,0]^{r[\Lambda]}=G[\mu,0,\Lambda]^{c[\Lambda]}G[\mu,1,\Lambda]$　/Fp　μ=0,...,n
$\Pi_{\mu=1}^{n+m} g[\mu,\Gamma]r[\mu]=\Pi_{\mu=1}^{n+m} g[\mu,\Gamma]^{\{\Sigma_{\nu=1}^{n+m'} A[\mu,\nu]c[\nu]\}}$ /Fp
$=\Pi_{\nu=1}^{n+m'}(\Pi_{\mu=1}^{n+m} g[\mu,\Gamma]^{A[\mu,\nu]})^{c[\nu]}$ /Fp
$=\Pi_{\nu=1}^{n+m'} g''[\nu,\Gamma]^{c[\nu]}$ /Fp
より分かる。

【０２７２】準元係数をコミットしたものと、これに付随する応答と準応答が検証式を満たすことは、
$u^{r'}=u^{\{\lambda[0]+\Sigma_{i=1}^{n}\lambda[i]r[i]r[i]\}}$ /Fp
$=u^{\lambda[0]}\Pi_{i=1}^{n}(u^{\lambda[k]})^{r[k]r[k]}$ /Fp
$=u[0]\Pi_{i=1}^{n}u[i]^{r[i]r[i]}$ /Fp　※
$v'^{r[0]}\Pi_{i=1}^{n}v^{r[i]r[i]}$ /Fp
$=(v^{r'[0]})^{r[0]}v^{\{\Sigma_{i=1}^{n}\Sigma_{\mu=0}^{n}\Sigma_{\nu=0}^{n}A[i,\mu]A[i,\nu]c[\mu]c[\nu]\}}$ /Fp
$=v^{\{r'[0]\Sigma_{\mu=0}^{n}[0,\mu]c[\mu]+2\Sigma_{i=1}^{n}\Sigma_{j=1}^{n}A[i,0]A[i,j]c[j]+\Sigma_{i=1}^{n}A[i,0]A[i,0]+\Sigma_{i=1}^{n}\Sigma_{j=1}^{n}\Sigma_{k=1}^{n}A[i,j]A[i,k]c[j]c[k]\}}$ /Fp
$=v^{\{\Sigma_{i=1}^{n}\phi[i]c[i]+\phi[0]+\Sigma_{i=1}^{n}c[i]c[i]\}}$ /Fp
$=\omega v^{\{\Sigma_{i=1}^{n}(c[i]c[i]+\phi[i]c[i])\}}$ /Fp
より成立することがわかる。以上、A[i,j]が置換行列であるという事実を使った。

【０２７５】前記実施例(2)の恒等式係数に関しては、
$\Sigma_{i=1}^{n}r[i]r[i]r[i]+\Sigma_{i=1}^{n}\rho''\lambda[i]r[i]r[i]+\rho'r[0]$ /Fq
$=\Sigma_{h=1}^{n}\Sigma_{i=1}^{n}\Sigma_{j=1}^{n}\Sigma_{k=1}^{n}A[h,i]A[h,j]A[h,k]c[i]c[j]c[k]$
$+\Sigma_{h=1}^{n}\Sigma_{i=1}^{n}\Sigma_{j=1}^{n}(3A[h,0]A[h,i]A[h,j]+\rho''\lambda[h]A[h,i]A[h,j])c[i]c[j]$
$+\Sigma_{h=1}^{n}\Sigma_{i=1}^{n}(3A[h,0]A[h,0]A[h,i]+2\rho''\lambda[h]A[h,0]A[h,i]+\rho'A[0,i])c[i]$
$+\Sigma_{h=1}^{n}(A[h,0]A[h,0]A[h,0]+\rho''\lambda[h]A[h,0]A[h,0])+\rho''\lambda[0]+\rho'A[0,0]$
/Fq
$=\Sigma_{h=1}^{n}(c[h]c[h]c[h]+\psi[h]c[h]c[h]+\phi[i]c[i]+\phi[0])$ /Fq
であり、これは、$v^{\{\Sigma_{h=1}^{n}(c[h]c[h]c[h]+\psi[h]c[h]c[h]+\phi[i]c[i])\}\omega[0]}$ /Fpの指数部と等しい。

【０２７６】以上で最後の式を導くためにA[i,j]が置換行列であるという事実を使った。

【０２７７】前記実施例(3)、及び実施例(4)に関しても同様の議論より分かる。

【０２７８】前記実施例(8)の証明付公開鍵列方法が出力した公開鍵列底と、公開鍵列対と疑似公開鍵列対と、これに付随する応答と挑戦値が検証処理の検証式を満たすことは、
$g'[\mu,0]^{r}=g'[\mu,0]^{cx'+a}$ /Fp
$=g'[\mu,0]^{cx}g'[\mu,0]^{a}$ /Fp
$=g'[\mu,1]^{c}g'[\mu,2]$ /Fp
＊を確認し、さらに部分復号が、このG[μ,0,Λ]を用いて行われたことを確認して受理する。

【０２６９】以上を、t人の証明者全てにより行われた結果を復号文とする。

【０２７０】［正当性］以上説明した実施例の正当性について説明する。

【０２７１】［完全性］入力文列と、出力暗号文列と変換情報保有コミットメントとである出力文列と、これに付随する応答と挑戦値が変換情報保有検証処理の検証式を満たすことは、
※よりわかる。

【０２７３】変換条件コミットメント生成処理が出力した恒等式の係数と、これに付随する応答と挑戦値が知識検証処理の検証式を満たすことは、以下のようにして分かる。

【０２７４】実施例(1)の恒等式係数に関しては、
★$v''^{r'}v'^{r[0]}\Pi_{i=1}^{n}v^{r[i]r[i]r[i]}$ /Fp
のvに対する指数部が、
$\Sigma_{i=1}^{n}r[i]r[i]r[i]+\Sigma_{i=1}^{n}\rho''\lambda[i]r[i]r[i]+\rho'r[0]$ /Fq
から分かる。

【０２７９】［健全性］与えられた挑戦値c[ν];ν=1,...,n+m'に対する、変換情報保有検証処理における検証式を満たす応答r[μ];μ=1,...,n+mを求めるにはA[μ ν];μ=1,...,n+m;ν=1,...,n+m'を知らなければならない。

【０２８０】これは、与えられた　g[μ,Γ]、g''[ν Γ];μ=1,...,n+m;ν=1,...,n+m'に対して、A[μ ν];μ=1,...,n+m;ν=1,...,n+m'を知らずして、等価検証処理における検証式を満たす応答を求めることは、離散対数問題を解くことに等しいからである。

【０２８１】なぜならば、A[μ,ν]を知らないと言う事は、少なくとも一つのg''[ν,Γ]に関しては、g[μ,

Γ]；μ=1,...,n+mを基底とするその表現を知らない。その時、任意のcに関して、検証式を満たす応答を求めることができるなら、c[ξ]=1、c[ν]=0；ν=0,...,ξ-1,ξ+1,...,n+m'なるc[ν]を選ぶ事で、離散対数を解くことが出来るからである。

【０２８２】また、挑戦値c[ν]は、コミットメントg[μ,Γ],g''[μ,Γ]を引数に持つため、挑戦値が決定してから、コミットメントを調節することができない(挑戦値生成関数がこの性質を持つことを要求する)。そのため、証明者には、挑戦値はコミットメント決定後に与えられた乱数と考えることができる。

【０２８３】どのg[μ,Γ]の成分に関しても、他の成分を基底とするその表現を知らなければ、検証式を満たす応答を複数作る事は、離散対数問題を解く事に等しい。なぜなら、異なるr[μ]、r'[μ]に関して、検証式が成り立つならば、両辺を互いに割る事により、g[μ,Γ]を基底とする、非自明な1の表現が得られる。これは離散対数問題を解く事に等しいからである。

【０２８４】入力文列生成方法により生成される入力文列　g[μΓ]；μ=1,...,n+m；Γ=0,...は、いずれかのΓに関してなすベクトル　g[μ,Γ]；μ=1,...,n+mが、ハッシュ関数で生成されているか、あるいはハッシュ関数で生成されているベクトルを乗じた等の操作で生成されていることが明らかになっているため、互いに、他を基底として表現する事が計算量的に困難である、と考えられる。

【０２８５】以上により、検証式を満たすr[μ]；μ=1,...,n+mとして、$g''[\nu,\Gamma]=\Pi_{\mu=1}^{n+m} g[\mu,\Gamma]^{A[\mu,\nu]}$ /Fp；ν=1,...,n+m'なるA[μ,ν]を用いて、$r[\mu]=\Sigma_{\nu=1}^{n+m'} A[\mu,\nu]c[\nu]$ /Fq；μ=1,...,n+m、と生成する以外には、証明者は計算できない。個別公開鍵を用いる方法でも同様である。

【０２８６】上述のようにあるΓに関して
$g''[\nu,\Gamma]=\Pi_{\mu=1}^{n+m} g[\mu,\Gamma]^{A[\mu,\nu]}$ /Fp　　ν=1,...,n+m'
の関係が証明されたならば、他のΓに関しても以下のように同様に証明される。

【０２８７】挑戦値生成関数の引数に含めたg[μ,Γ],g''[ν,Γ]に対して検証式が成り立つならば、
$g''[\nu,\Gamma]=\Pi_{\mu=1}^{n+m} g[\mu,\Gamma]^{A[\mu,\nu]}$ /Fp　　ν=1,...,n+m'
である。

【０２８８】なぜならば、

＊　$g''[\nu,\Gamma]=\Pi_{\mu=1}^{n+m} g[\mu,\Gamma]^{A'[\mu\nu]}$ /Fp　　ν=1,...,n+m'
と表した時に検証式が成り立つならば、
$=\Pi_{\mu=1}^{n+m} g[\mu,\Gamma]^{\{\Sigma_{\nu=1}^{n+m'}(A[\mu,\nu]-A'[\mu,\nu])c[\nu]\}}=1$ /Fp
が成り立つ。

【０２８９】ところが、無作為に選ばれたc[ν]に関して、これが成り立つのは、
$\Pi_{\mu=1}^{n+m} g[\mu,\Gamma]^{A[\mu,\nu]}=\Pi_{\mu=1}^{n+m} g[\mu,\Gamma]^{A'[\mu,\nu]}$ /Fp　　ν=1,...,n+m'
の時だけであるからである。

【０２９０】前記実施例(2)において、変換条件コミットメント生成処理により準元係数をコミットしたものu、u[μ]；μ=0,...,nが与えられたとき、応答r[i]；i=1,...,nと、準応答r'が検証式を満たす時、準応答r'は一意であり、
$r'=\lambda[0]+\Sigma_{i=1}^{n}\lambda[i]r[i]r[i]$ /Fq
が、検証式を満たすことよりr'は上式で表されるものである。

【０２９１】前記実施例(2)の恒等式の検証式の左辺のvの指数部を展開すると、
$\Sigma_{h=1}^{n}\Sigma_{i=1}^{n}\Sigma_{j=1}^{n}\Sigma_{k=1}^{n}A[h,i]A[h,j]A[h,k]c[i]c[j]c[k]+\Sigma_{h=1}^{n}\Sigma_{i=1}^{n}\Sigma_{j=1}^{n}(3A[h,0]A[h,i]A[h,j]+\rho''\lambda[h]A[h,i]A[h,j])c[i]c[j]+\Sigma_{i=1}^{n}(\Sigma_{h=1}^{n}(3A[h,0]A[h,0]A[h,i]+2\rho''\lambda[h]A[h,0]A[h,i])+\rho'A[0,i])c[i]+\Sigma_{h=1}^{n}(A[h,0]A[h,0]A[h,0]+\rho''\lambda[h]A[h,0]A[h,0])+\rho''\lambda[0]+\rho'A[0,0]$ /Fq
である。

【０２９２】右辺のvの指数部は、
$\Sigma_{i=1}^{n}(c[i]c[i]c[i]+\psi[i]c[i]c[i]+\phi[i]c[i])+\phi[0]$ /Fq
である。

【０２９３】よって、いかなるc[μ]；μ=0,...,nに関しても、検証式が成り立つためには、c[μ]c[ν]c[ξ]；μ,ν,ξ=0,...,nの係数が同じでなければならない。それ以外の時に、検証式が無作為に与えられたc[μ]に関して成り立つ可能性は、無視できる。

【０２９４】これは、
δ[i,j]=1　　i=jの場合
=0　　その他
δ'[i,j,k]=1　　i=j=kの場合
=0　　その他
＊　を用いて、

$\Sigma_{h=1}^{n}A[h,i]A[h,j]A[h,k]=\delta'[i,j,k]$ /Fq　　i,j,k=1,...,n

$\Sigma_{h=1}^{n}(3A[h,0]A[h,i]A[h,j]+\rho''\lambda[h]A[h,i]A[h,j])=\delta[i,j]\psi[i]$ /Fq　　i,j=1,...,n

$\Sigma_{h=1}^{n}(3A[h,0]A[h,0]A[h,i]+2\rho''\lambda[h]A[h,0]A[h,i])+\rho'A[0,i]=\phi[i]$ /Fq　　i=1,...,n

$\Sigma_{h=1}^{n}(A[h,0]A[h,0]A[h,0]+\rho''\lambda[h]A[h,0]A[h,0])+\rho''\lambda[0]+\rho'A[0,0]=\phi[0]$ /Fq

であることを保証する。このことから、A[i,j];i,j=1,...,nについて以下のことが分かる。

【０２９５】与えられたj,k;j≠kに対して、第h成分がA[h,j]A[h,k]であるn次元ベクトルA[h,j]A[h,k];h=1,...,nと、与えられたiに対して第h成分がA[h,i]であるn次元ベクトルA[h,i];h=1,...,nとを考える。今、n個のベクトルA[h,i];i=1,...,nがn次元空間を張る、すなわち全てのベクトルがA[h,i];i=1,...,nの線形結合で表せるとする。すると、上式よりベクトルA[h,j]A[h,k];h=1,...,nは、すべてのベクトルA[h,i]と内積が0であるので、

$A[h,j]A[h,k]=0 \quad /F_q \qquad h=1,\ldots,n$

である。

【０２９６】このことから、n個のベクトルA[h,i];h=1,...,n;i=1,...,nのうち、各h成分が0でないベクトルはひとつしかない。

【０２９７】また上式より、i=j=kの時、A[h,i]A[h,j]A[h,k]≠0であるため、ベクトルA[h,i];h=1,...,nは少なくともひとつは0でない成分を持つ。よって、全てのベクトルA[h,i];h=1,...,nは、0でない成分をただひとつ持ち、上式より、それは $1^{1/3}$ である。

【０２９８】今度は、n個のベクトルA[h,i];h=1,...,n;i=1,...,nがｎ次元空間を張ることを示す。

【０２９９】ベクトルa[h];h=1,...,nを、n個のスカラーκ[i];i=1,...,nを用いて、

$a[h]=\Sigma_{i=1}{}^{n}\ \kappa[i]A[h,i] \qquad h=1,\ldots,n \quad /F_q$

と表す。

【０３００】もし、a[h]= $/F_q$ ならば、κ[i]=0であることを示せば、n個のベクトルA[h,i];h=1,...,n;i=1,...,nがn次元空間を張ることを示せる。a[h]=0 $/F_q$ ならば、上式の両辺に、第h成分がA[h,i]A[h,i]であるn次元ベクトルA[h,i]A[h,i]を掛けると、上式の二つ上の式より、

$0=\kappa[i] \quad /F_q \qquad i=1,\ldots,n$

となる。以上をもってA[i,j]が置換行列であるか、置換行列のいくつかの成分に $1^{1/3}$ を乗じることによって得られる準置換行列であることが示された。

【０３０１】実施例(1)の恒等式の検証式の左辺のvの指数部を展開すると、

$r[0]r[0]+\Sigma_{i=1}{}^{n}r[i]r[i] \quad /F_q$

$=\Sigma_{i=1}{}^{n}\Sigma_{j=1}{}^{n}\Sigma_{k=1}{}^{n}A[i,j]A[i,k]c[j]c[k]$

$+\Sigma_{j=1}{}^{n}(\Sigma_{i=1}{}^{n}2A[i,0]A[i,j]+r'[0]A[0,j])c[j]$

$+\Sigma_{i=1}{}^{n}A[i,0]A[i,0]+r'[0]A[0,0] \quad /F_q$

である。右辺のvの指数部は、

$\Sigma_{i=1}{}^{n}(c[i]c[i]+\phi[i]c[i])+\phi[0] \quad /F_q$

である。

【０３０２】よっていかなるc[μ];μ=0,...,nに関しても、検証式が成り立つためには、c[μ]cν;μ,ν=0,...,nの係数が同じでなければならない。それ以外の時に検証式が無作為に与えられた応答に関して成り立つ可能性は無視できる。

【０３０３】これは

$\Sigma_{h=1}{}^{n}A[h,i]A[h,j]=\delta[i,j] \quad /F_q$

$\phi[i]=\Sigma_{h=1}{}^{n}2A[h,0]A[h,i]+r'[0]A[0,i] \quad /F_q$

$\phi[0]=\Sigma_{i=1}{}^{n}A[i,0]A[i,0]+r'[0]A[0,0] \quad /F_q$

であることを保証する。よって、A[i,j];i,j=1,...,nは正規直交行列でなければ検証式は成り立つ可能性は無視できる。

【０３０４】前記実施例(3)、実施例(4)においても同様の議論が成り立ち、A[i,j];i,j=1,...,nは、置換行列かつ正規直交行列である。そして、これは置換行列であることを意味する。

【０３０５】［知識隠匿性］再暗号シャッフル証明文において、再暗号シャッフルの情報が計算量的に秘匿されていることを示す。

【０３０６】再暗号シャッフルの結果g''[ν],m''[μ]以外にも、

r[μ],r',φ[i],ψ[i],ω,v',v'',v,u,u[i],

$\underline{r}[\mu],\underline{\phi}[i],\underline{v'},\underline{\omega},v$

等の値が明らかになる。これらは再暗号シャッフルに関する情報を与えている。しかし指数演算された結果でない条件の数より、再暗号シャッフル行列演算に関する未知数のほうが多くなるように、恒等式の係数をコミットして隠蔽すれば離散対数問題をといて条件の数を増やさない限り解けない。ただし変数の数だけでなく未知数の条件の中への現れ方で解ける場合もあるので、若干の調整を行う必要がある。

【０３０７】

【発明の効果】以上説明したように、本発明によれば、証明付再暗号シャッフルの計算量を、従来の技術と比べて、低減する、という効果を奏する。

【０３０８】特に、検証処理は事前に計算しておくことができない応用例が多いと考えられるので、検証の計算量を比較すると、入力暗号文の数をnとしたとき、従来の技術(1)では、安全変数が160の時冪乗剰余演算が320n+2n回必要とされており、従来の技術(2)では冪乗剰余演算が8(n log n −n+1)回必要とされていたのに対して、本発明によれば、冪乗剰余演算は7n+14回で済み、n>4の時は、いずれの従来の技術よりも冪乗剰余演算が少ない。

【０３０９】しかも、本発明においては、検証過程で行う冪乗剰余は、個別の冪乗剰余演算ではなく、冪乗剰余演算の積の計算であることから、個別の冪乗剰余演算よりも少ない計算量で計算できるため、更なる高速化が望める、という効果を奏する。

【図面の簡単な説明】

【図１】従来の技術１の構成を示す図である。

【図２】従来の技術２の構成を示す図である。

【図３】本発明の実施例における証明付再暗号シャッフル装置と再暗号シャッフル検証装置との情報の入出力を

示す図である。
【図４】本発明の実施例１の証明付再暗号シャッフル装置の詳細を示す図である。
【図５】本発明の実施例１の再暗号シャッフル検証装置の詳細を示す図である。
【図６】本発明の実施例２の証明付再暗号シャッフル装置の詳細を示す図である。
【図７】本発明の実施例２の再暗号シャッフル検証装置の詳細を示す図である。
【図８】本発明の実施例３の証明付再暗号シャッフル装置の詳細を示す図である。
【図９】本発明の実施例３の再暗号シャッフル検証装置の詳細を示す図である。
【図１０】本発明の実施例４の証明付再暗号シャッフル装置の詳細を示す図である。
【図１１】本発明の実施例４の再暗号シャッフル検証装置の詳細を示す図である。
【図１２】本発明の実施例５の入力文列生成装置の詳細を示す図である。
【図１３】本発明の実施例６の入力文列生成方法の情報の入出力を示す図である。
【図１４】本発明の実施例６における前処理装置の詳細を示す図である。
【図１５】本発明の実施例７の入力文列生成方法の情報の入出力を示す図である。
【図１６】本発明の実施例７の入力文列生成方法の情報の入出力を示す図である。
【図１７】本発明の実施例７における証明付暗号化装置の詳細を示す図である。
【図１８】本発明の実施例７における暗号化検証装置の詳細を示す図である。
【図１９】本発明の実施例６と実施例７における証明付公開鍵列生成装置の詳細を示す図である。
【図２０】本発明の実施例６と実施例７における公開鍵列検証装置の詳細を示す図である。
【符号の説明】
１００　入力暗号文列
１０１　暗号シャッフル
１０２　出力暗号文列
１０３　疑似出力暗号文列
１０４　挑戦値生成関数
１０５　挑戦値
１０６　応答
２００　置換
３００　入力文列
３０１　入力暗号文列
３０３　公開鍵
３０３　再暗号シャッフル情報
３０４　再暗号シャッフル行列
３０５　再暗号秘密乱数
３０６　情報隠蔽
３０７　シャッフル行列
３０８　元係数
３０９　準元係数
３１０　係数基底
３１１　その他
３１２　証明付き再暗号シャッフル行列装置
３１３　出力暗号文列
３１４　暗号シャッフル証明文
３１５　変換情報保有コミットメント
３１６　変換条件コミットメント
３１７　応答
３１８　準応答
３１９　再暗号シャッフル検証装置
４００、６００、８００、１０００　入力文列
４０、６０１、８０１、１００６　再暗号シャッフル情報
４０２、６０２、８０２、１００１　再暗号シャッフル行列
４０３、６０３、８０３、８０５、１００２、１００５　元係数
４０４、６０４、６０５、８０６、１００３　係数基底
４０５、６０２、８０８、１００９　再暗号シャッフル行列作用
４０６、６０３、８０９、１０１０　出力文列
４０７、６０４、８１０、１０１１　出力暗号文列
４０８、６０５、８１１、１０１２　保有コミット
４０９、６０６、８１２、８１６、１０１３、１０２２　恒等式係数計算
４０Ａ、６１４、８２３、１０１７　コミットメント
４１０、６０７、８１３、８１７、１０１４、１０２３　恒等式係数
４１１、６０８、８１４、８１８、１０１５、１０２４　隠蔽処理
４１２、６１３、８１５、８２２、１０１６、１０２８　条件コミット
４１３、６１５、８２４、１０３０、１０３４　挑戦値生成
４１４、６１６、８２５、１０３１、１０３５　挑戦値
４１５、６１７、８２６、１０３２、１０３６　応答生成
４１６、６１８、８２７、１０３３、１０３７　応答
４１７、８１０、１０１１　出力暗号文列
４１８、６２２、８３１、１０４０　再暗号シャッフル証明文
４１９、６２３、８３２、１０４２　変数情報保有コミットメント生成処理
４２０、６２５、８３４、１０４３　変換条件コミットメント生成処理
４２１、６２４、８３５、１０４６、１０４７　応答生

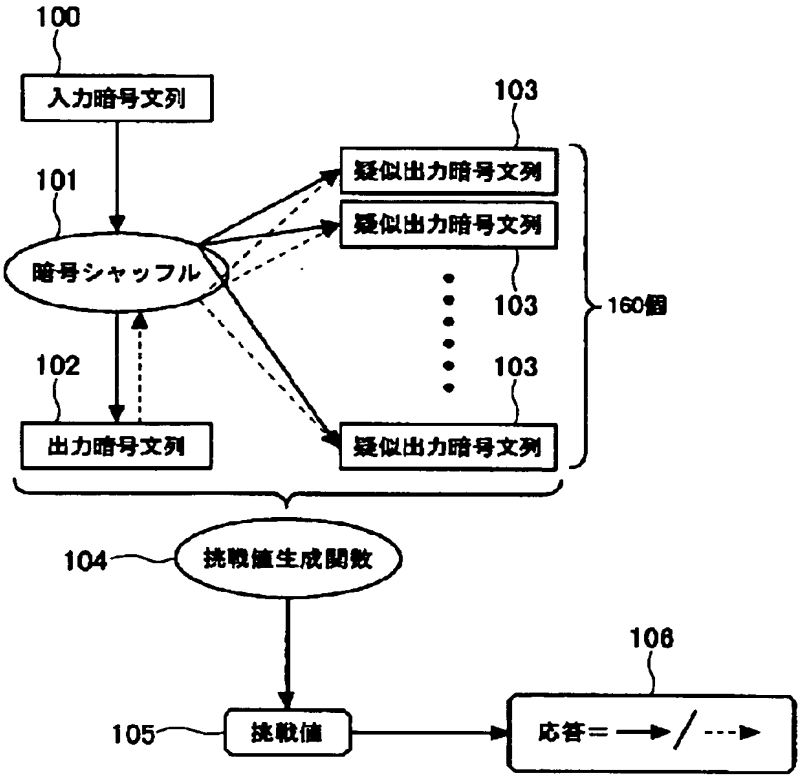
59

成処理
５００　挑戦値生成関数
５０１　挑戦値
５０２、７０６、９０２、１１０３、１１０５　保有検証
５０３　条件検証
５０４、７０９、９０６、１１１０　検証結果
５０５、７１０　　変換情報保有検証処理
５０６、７１１　変換条件検証処理
６１０、１０２６　準元係数隠蔽処理
６１２、８２１、１０２７　準応答コミット
６１９、８２８　準応答生成
６２０、８２９　準応答
６０Ａ、８０７、１００７　準元係数
７０４、９００、１１００　挑戦値生成
７０５、９０１、１１０１　挑戦値
７０７、９０３、９０４、１１０２　恒等式検証
７０８、９０５、１１０７　準応答検証
８２０　準元係数隠蔽処理
８２１　準応答コミット
１００４　情報隠蔽因子
１０１８　選択処理
１０１９　入力文列
１０２０　保有生成
１０２１　保有コミット
１１１１、１１１４　条件検証処理
１１１２、１１１３　保有検証処理
１２００、１４００　基底生成関数
１２０１　入力ベクトル
１３００、１５０１、１９０１　公開鍵列情報
１３０１、１５０２、１９０２　秘密鍵
１３０２、１５０３、１９０３　疑似秘密鍵
１３０４、１５０４　証明付き公開鍵列装置
１３０５、１３０６、１５０５、１５０６　分散公開鍵列対
１３０９　前処理装置
１３０８　検証結果
１４０１、１４０３　公開鍵列基底
１４０２　前処理
１４０４　公開鍵列

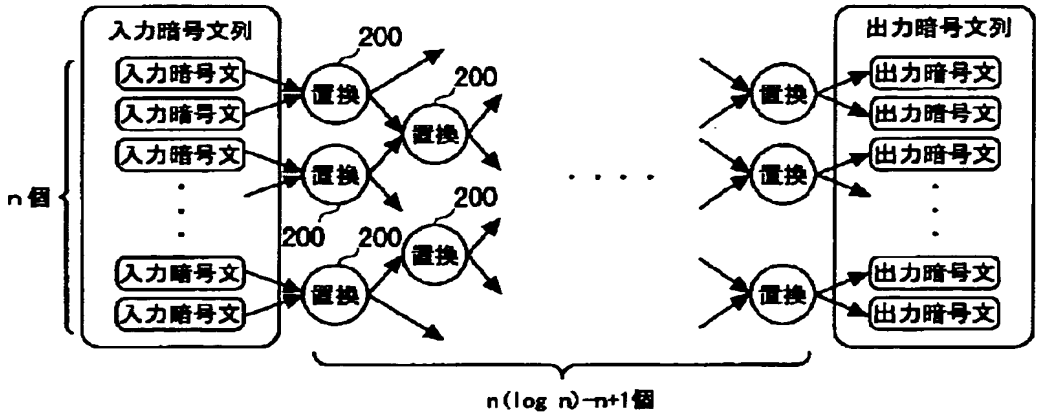
60

１５００　共通初期値
１５０７　公開鍵列検証装置
１５０８　検証結果
１５０Ａ　共通初期値
１６００　共通公開鍵
１６０１　個別公開鍵
１６０２　平文
１６０３　暗号化技術
１６０４　秘密乱数
１６０５　疑似秘密乱数
１６０５　証明付き暗号化装置
１６０７　入力暗号文
１６０８　暗号化証明文
１６０９　暗号化検証装置
１６１０、１８１０　検証結果
１７００、１７０１、１７０２　べき剰余演算
１７０３　暗号文底
１７０４　コミットメント
１７０６、１９１１　挑戦値生成
１７０７、１９１２　挑戦値
１７０８、１９１３　応答生成
１７０９、１９１４　応答
１７１０、１９０４　基底生成関数
１７１１　公開鍵基底
１７１２　平文
１７１３　暗号文対
１８００、２０００　挑戦値生成関数
１８０１、２００３　挑戦値
１８０２　暗号化検証
１９００　　初期値
１９０４、２０００　基底生成関数
１９０５、２００２　公開鍵列底
１９０６　公開鍵列対生成
１９０７　公開鍵列対
１９０８　疑似公開鍵列対生成
１９１５　公開鍵列証明文
１９１７　公開鍵列対
２００４　公開鍵列検証
２００５　検証結果

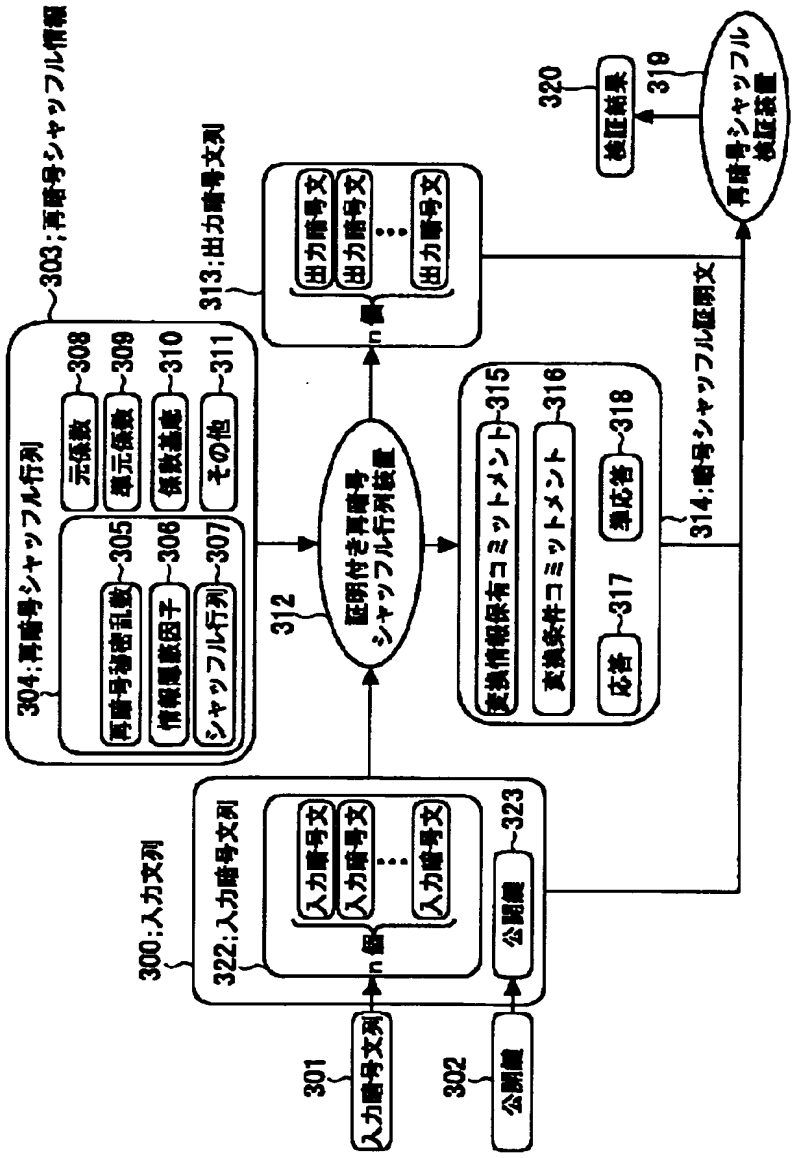
【図１】


100
入力暗号文列
101
暗号シャッフル
102
出力暗号文列
103
疑似出力暗号文列
疑似出力暗号文列
103
103
疑似出力暗号文列
160個
104
挑戦値生成関数
105
挑戦値
106
応答＝

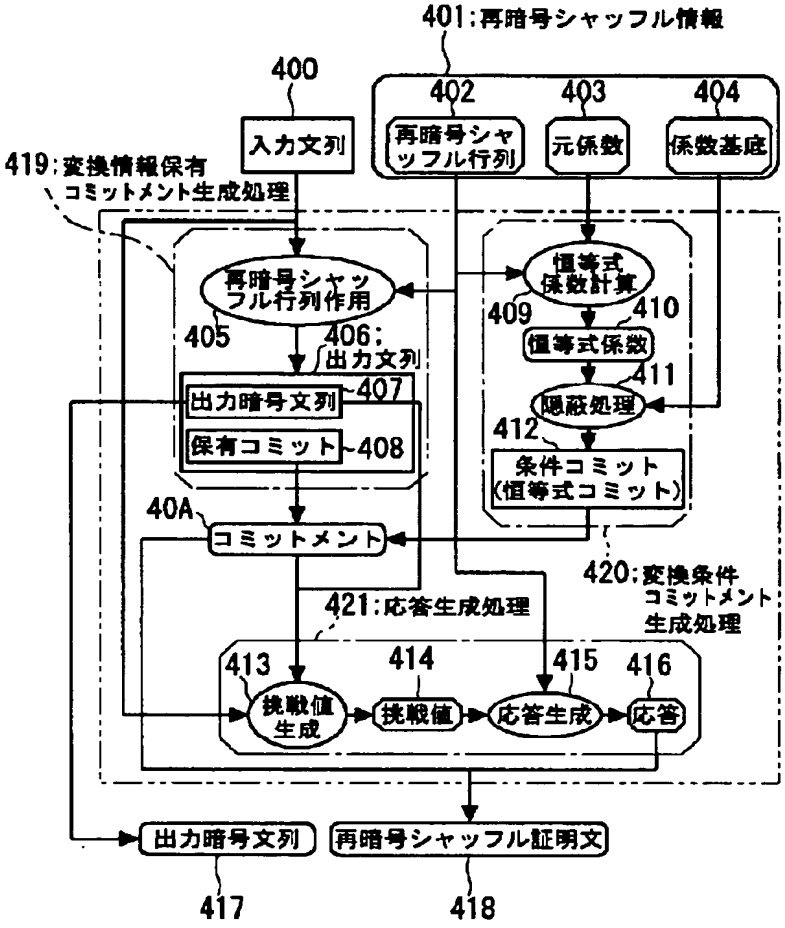
【図２】


入力暗号文列
入力暗号文
入力暗号文
入力暗号文
入力暗号文
入力暗号文
n個
置換
200
出力暗号文列
出力暗号文
出力暗号文
出力暗号文
出力暗号文
出力暗号文
n(log n)−n+1個

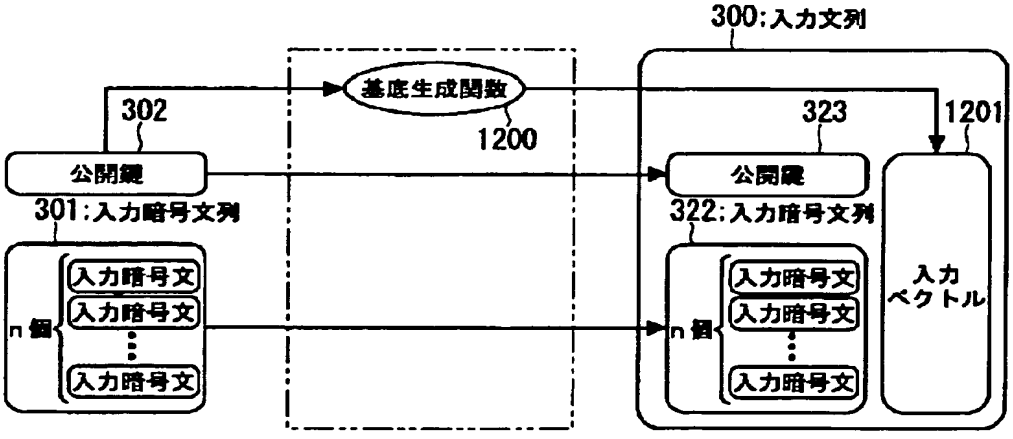
【図３】


301
入力暗号文列
302
公開鍵
300:入力文列
322:入力暗号文列
入力暗号文
入力暗号文
入力暗号文
n 個
323
公開鍵
312
証明付き再暗号
シャッフル行列装置
303:再暗号シャッフル情報
304:再暗号シャッフル行列
再暗号準備乱数
305
情報理論因子
306
シャッフル行列
307
元係数
308
単元係数
309
係数基底
310
その他
311
313:出力暗号文列
出力暗号文
出力暗号文
出力暗号文
n 個
314:暗号シャッフル証明文
変換情報保有コミットメント
315
変換条件コミットメント
316
単応答
318
応答
317
320
検証結果
319
再暗号シャッフル
検証装置

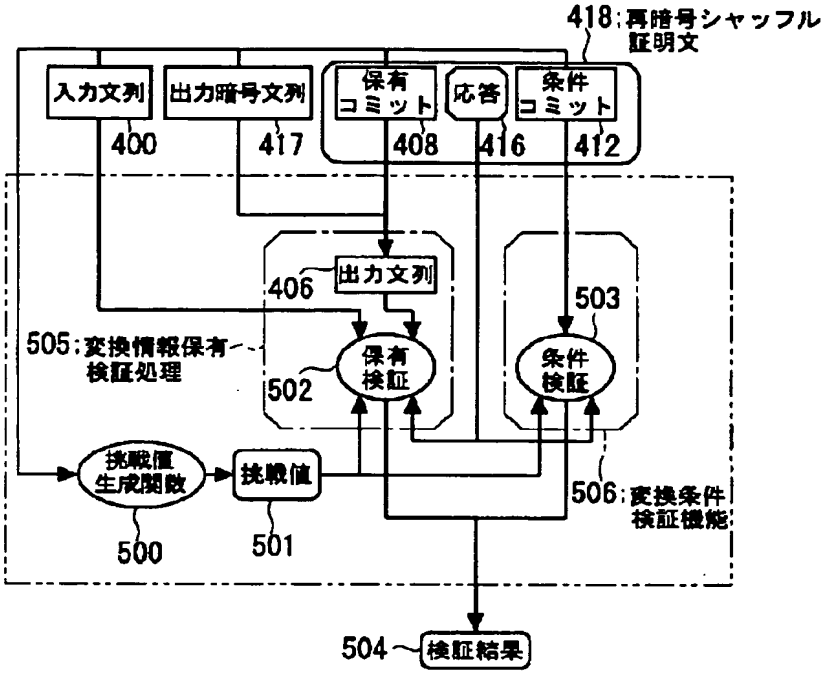
【図４】


401:再暗号シャッフル情報
400
402
403
404
入力文列
再暗号シャッフル行列
元係数
係数基底
419:変換情報保有コミットメント生成処理
再暗号シャッフル行列作用
恒等式係数計算
409
410
405
406:出力文列
恒等式係数
411
出力暗号文列
407
隠蔽処理
保有コミット
408
412
条件コミット(恒等式コミット)
40A
コミットメント
420:変換条件コミットメント生成処理
421:応答生成処理
413
414
415
416
挑戦値生成
挑戦値
応答生成
応答
出力暗号文列
再暗号シャッフル証明文
417
418

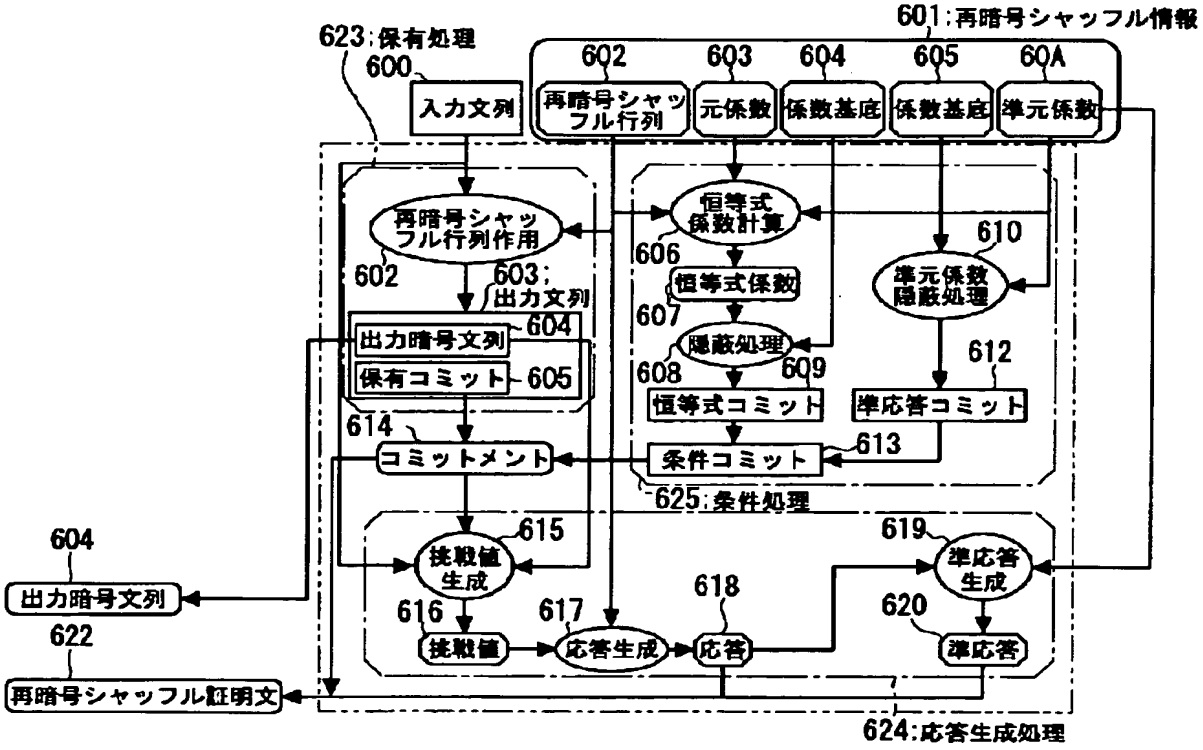
【図１２】


300:入力文列
基底生成関数
1200
302
323
1201
公開鍵
公開鍵
301:入力暗号文列
322:入力暗号文列
n個
入力暗号文
入力暗号文
入力暗号文
n個
入力暗号文
入力暗号文
入力暗号文
入力ベクトル

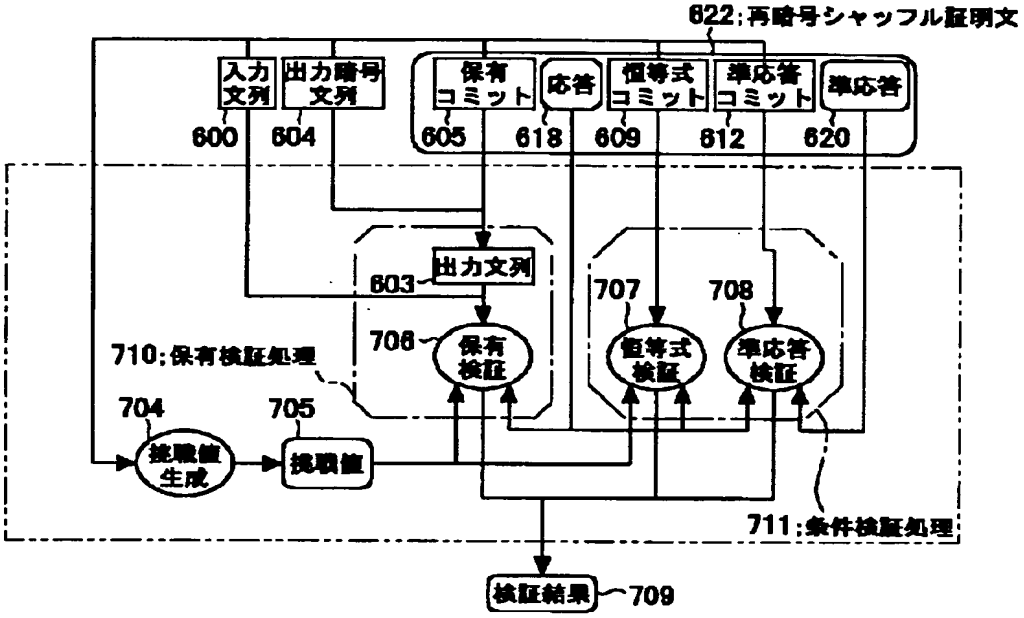
【図５】


418:再暗号シャッフル証明文
入力文列
出力暗号文列
保有コミット
応答
条件コミット
400
417
408
416
412
406
出力文列
503
505:変換情報保有検証処理
保有検証
502
条件検証
挑戦値生成関数
挑戦値
506:変換条件検証機能
500
501
504
検証結果

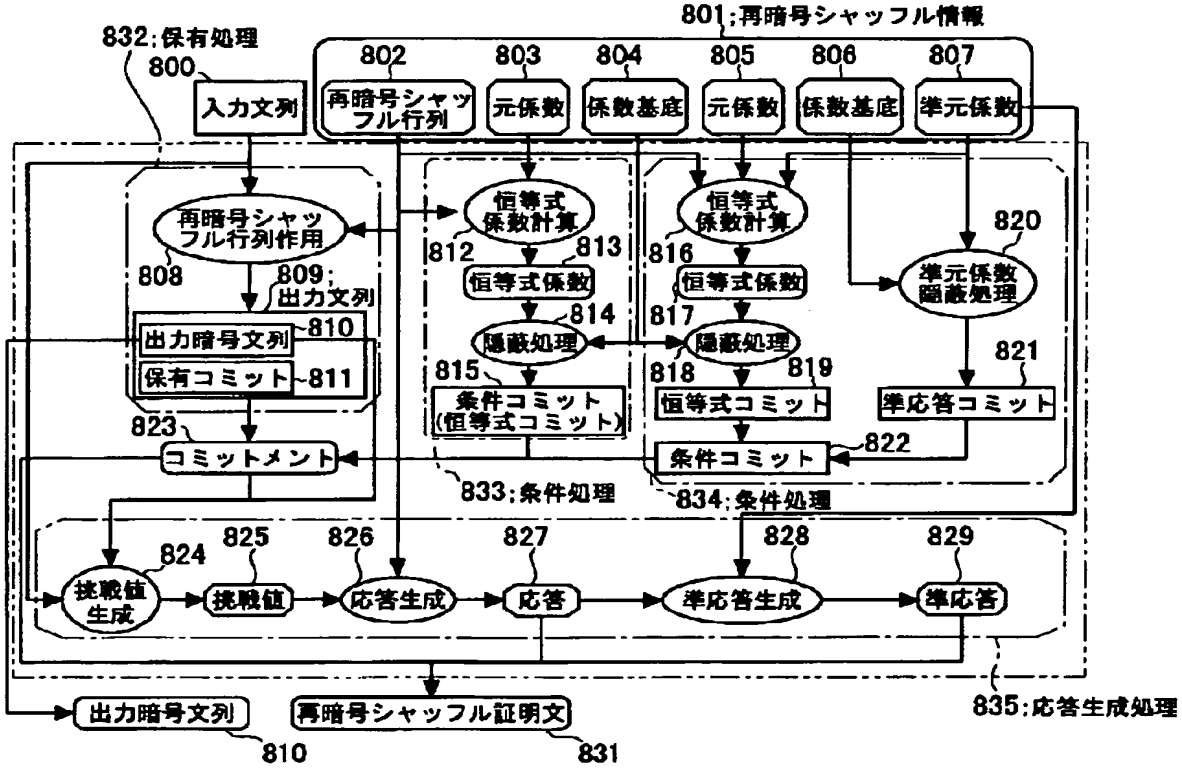
【図６】


601:再暗号シャッフル情報
623:保有処理
600
602
603
604
605
60A
入力文列
再暗号シャッフル行列
元係数
係数基底
係数基底
準元係数
再暗号シャッフル行列作用
恒等式係数計算
610
606
602
603:出力文列
恒等式係数
準元係数隠蔽処理
607
604
出力暗号文列
隠蔽処理
612
608
609
保有コミット
605
恒等式コミット
準応答コミット
614
613
コミットメント
条件コミット
625:条件処理
615
619
604
挑戦値生成
準応答生成
618
616
617
620
出力暗号文列
622
挑戦値
応答生成
応答
準応答
再暗号シャッフル証明文
624:応答生成処理

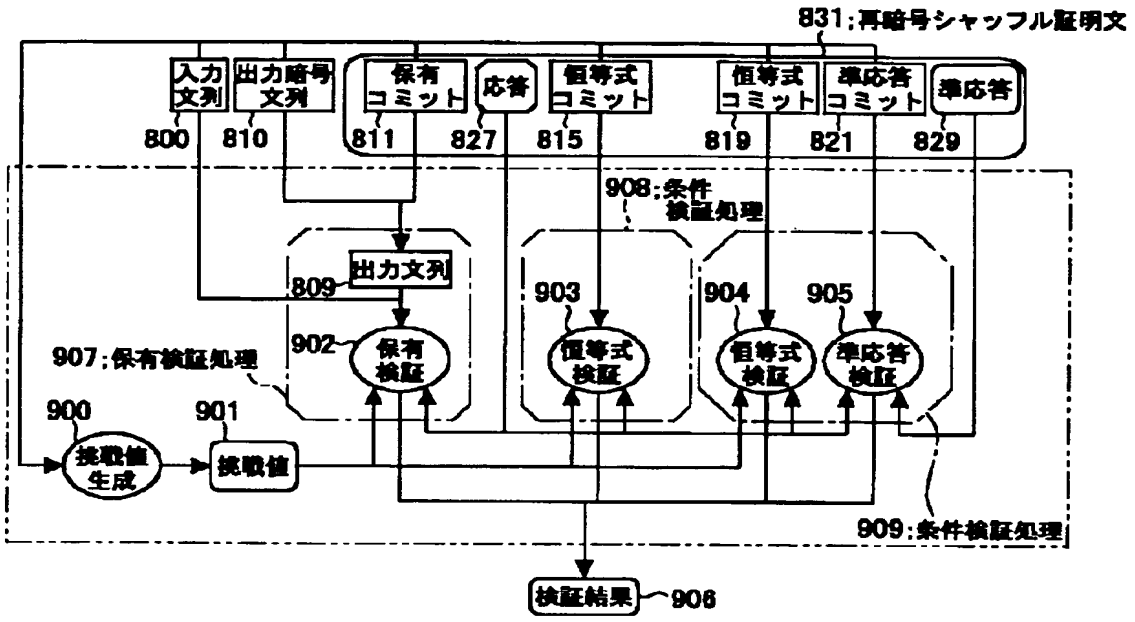
【図７】


622:再暗号シャッフル証明文
入力文列
出力暗号文列
保有コミット
応答
恒等式コミット
準応答コミット
準応答
600
604
605
618
609
612
620
出力文列
603
707
708
706
保有検証
恒等式検証
準応答検証
710:保有検証処理
704
705
挑戦値生成
挑戦値
711:条件検証処理
検証結果
709

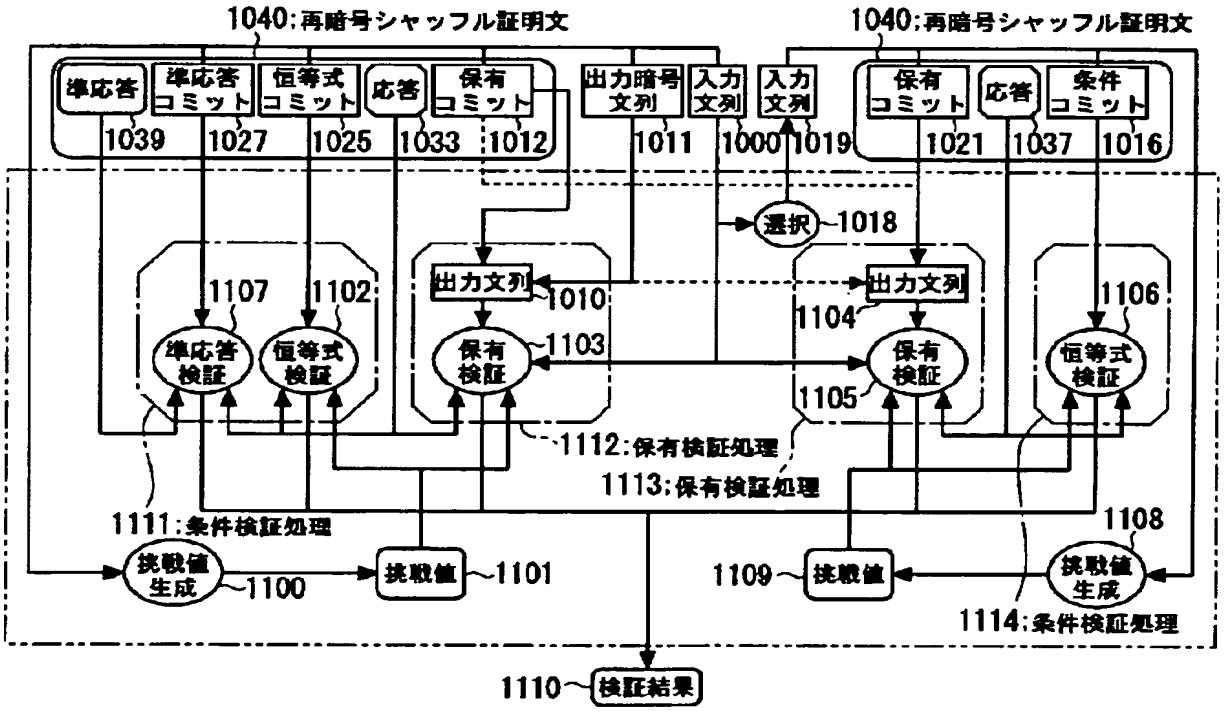
【図８】


801:再暗号シャッフル情報
832:保有処理
800
802
803
804
805
806
807
入力文列
再暗号シャッフル行列
元係数
係数基底
元係数
係数基底
準元係数
再暗号シャッフル行列作用
恒等式係数計算
恒等式係数計算
820
808
812
813
816
809:出力文列
恒等式係数
恒等式係数
準元係数隠蔽処理
出力暗号文列
810
814
817
隠蔽処理
隠蔽処理
821
保有コミット
811
815
818
819
条件コミット(恒等式コミット)
恒等式コミット
準応答コミット
823
822
コミットメント
条件コミット
833:条件処理
834:条件処理
824
825
826
827
828
829
挑戦値生成
挑戦値
応答生成
応答
準応答生成
準応答
835:応答生成処理
出力暗号文列
再暗号シャッフル証明文
810
831

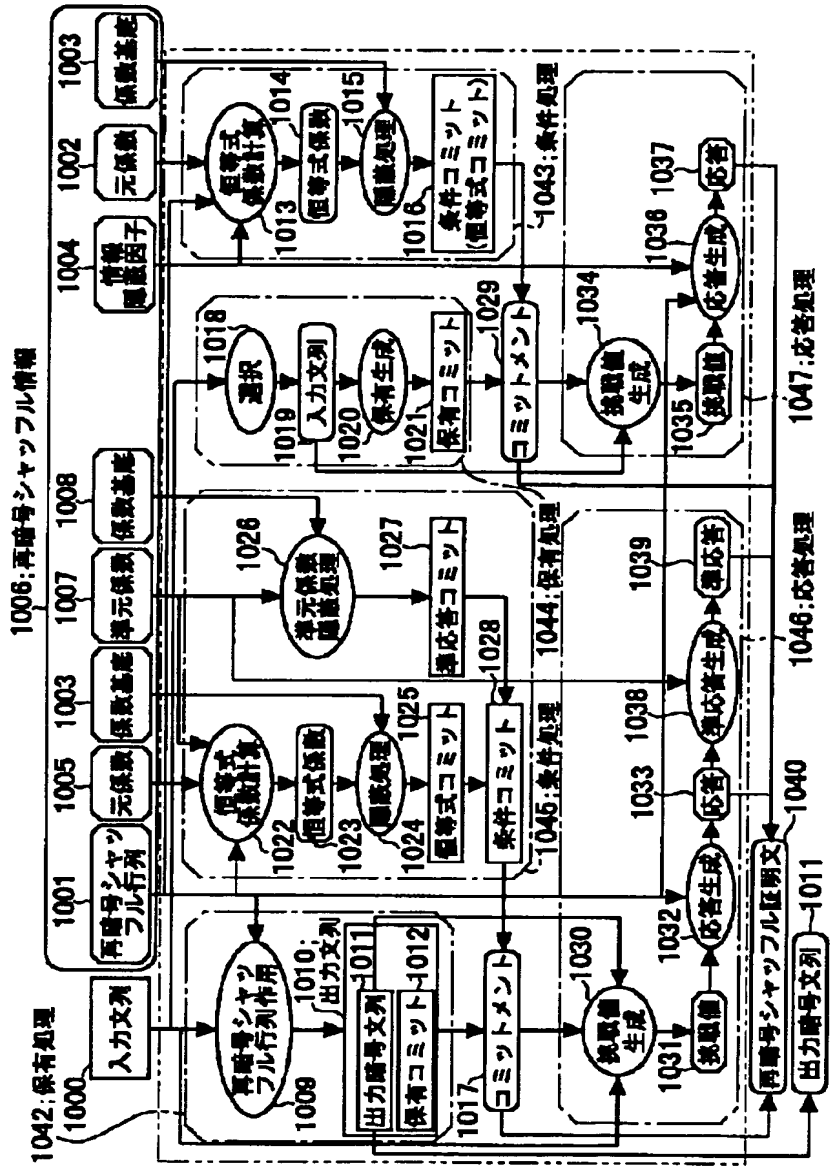
【図９】


831:再暗号シャッフル証明文
入力文列
出力暗号文列
保有コミット
応答
恒等式コミット
恒等式コミット
準応答コミット
準応答
800
810
811
827
815
819
821
829
908:条件検証処理
出力文列
809
902
保有検証
903
恒等式検証
904
恒等式検証
905
準応答検証
907:保有検証処理
900
901
挑戦値生成
挑戦値
909:条件検証処理
検証結果
906

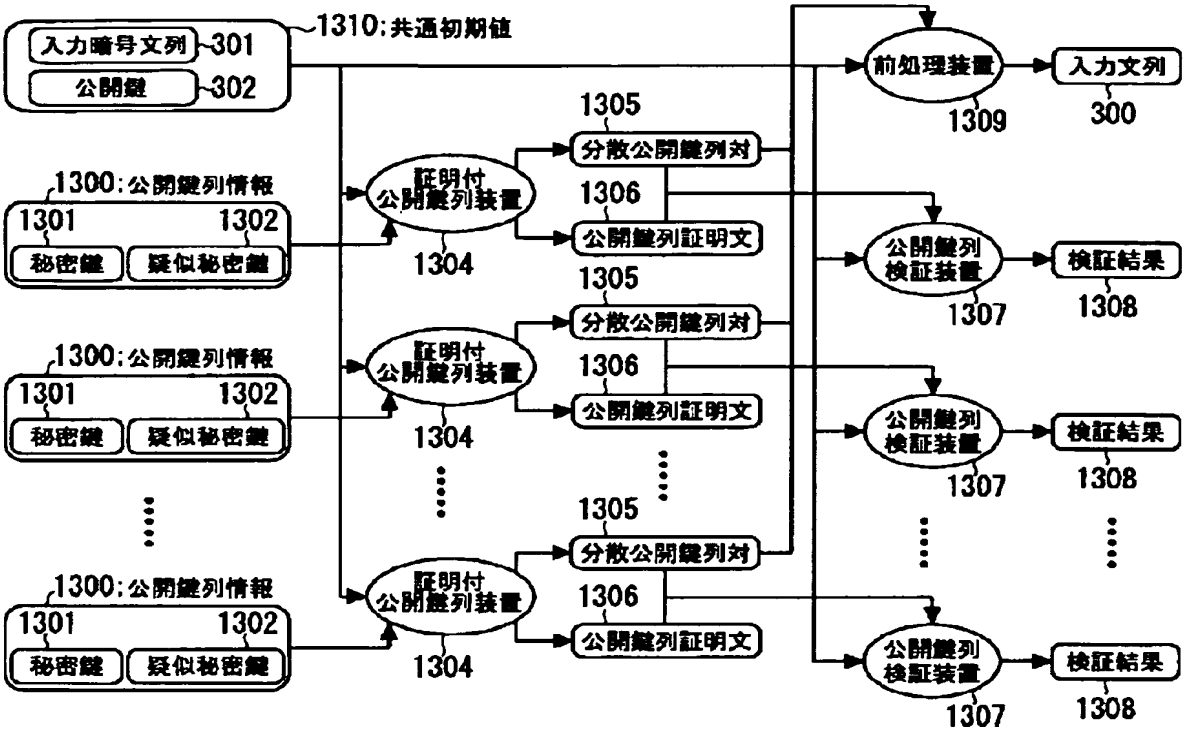
【図１１】


1040:再暗号シャッフル証明文
1040:再暗号シャッフル証明文
準応答
準応答コミット
恒等式コミット
応答
保有コミット
出力暗号文列
入力文列
入力文列
保有コミット
応答
条件コミット
1039
1027
1025
1033
1012
1011
1000
1019
1021
1037
1016
選択
1018
出力文列
1010
出力文列
1104
1107
1102
1106
準応答検証
恒等式検証
保有検証
1103
保有検証
恒等式検証
1105
1112:保有検証処理
1113:保有検証処理
1111:条件検証処理
1108
挑戦値生成
1100
挑戦値
1101
1109
挑戦値
挑戦値生成
1114:条件検証処理
1110
検証結果

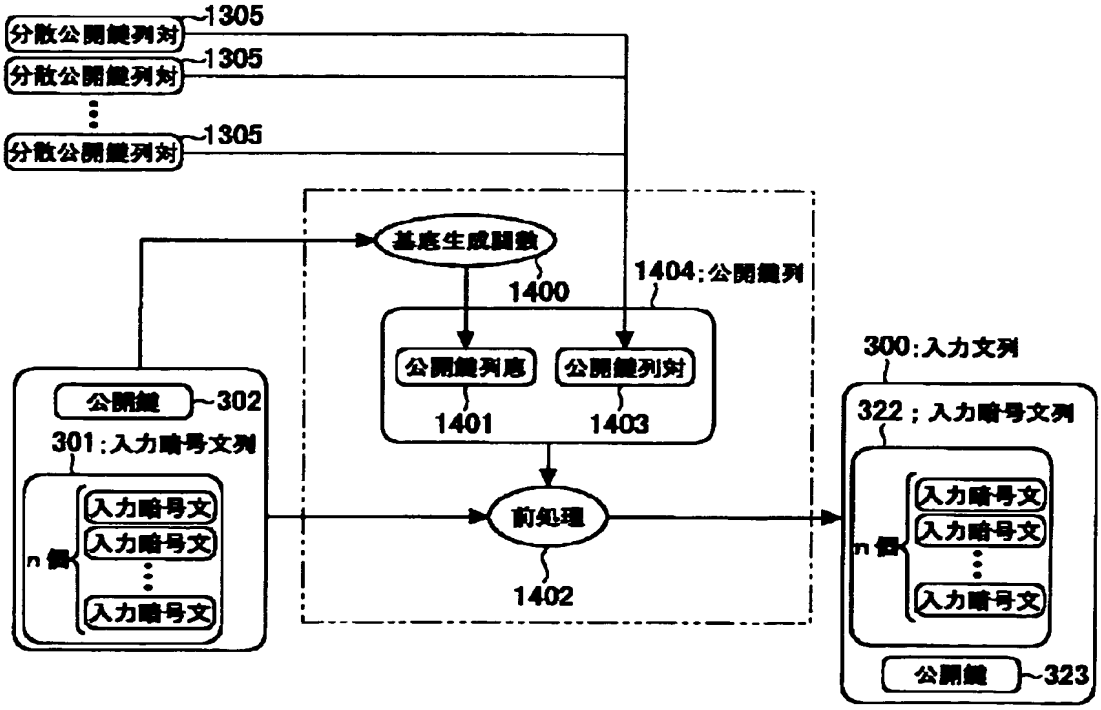
【図１０】


1042:保有処理
1000
入力文列
1001
再帰号シャッフル行列
1002
元係数
1003
係数基底
1004
情報源因子
1005
元係数
1006:再帰号シャッフル情報
1007
源元係数
1008
係数基底
再帰号シャッフル行列作用
1009
1010:出力文列
出力番号文列
1011
保有コミット
1012
コミットメント
1017
1013
冪等式係数計算
1014
冪等式係数
1015
冪等処理
1016
条件コミット(冪等式コミット)
1043:条件処理
1018
選択
1019
入力文列
1020
保有生成
1021
保有コミット
1029
コミットメント
1034
挑戦値生成
1035
挑戦値
1036
応答生成
1037
応答
1047:応答処理
1022
冪等式係数計算
1023
冪等式係数
1024
冪等処理
1025
冪等式コミット
1046:条件処理
1026
源元係数冪等処理
1027
源応答コミット
1028
条件コミット
1044:保有処理
1039
源応答
1038
源応答生成
1033
応答
1040
1030
挑戦値生成
1031
挑戦値
1032
応答生成
再帰号シャッフル反映文
出力番号文列
1011
1046:応答処理

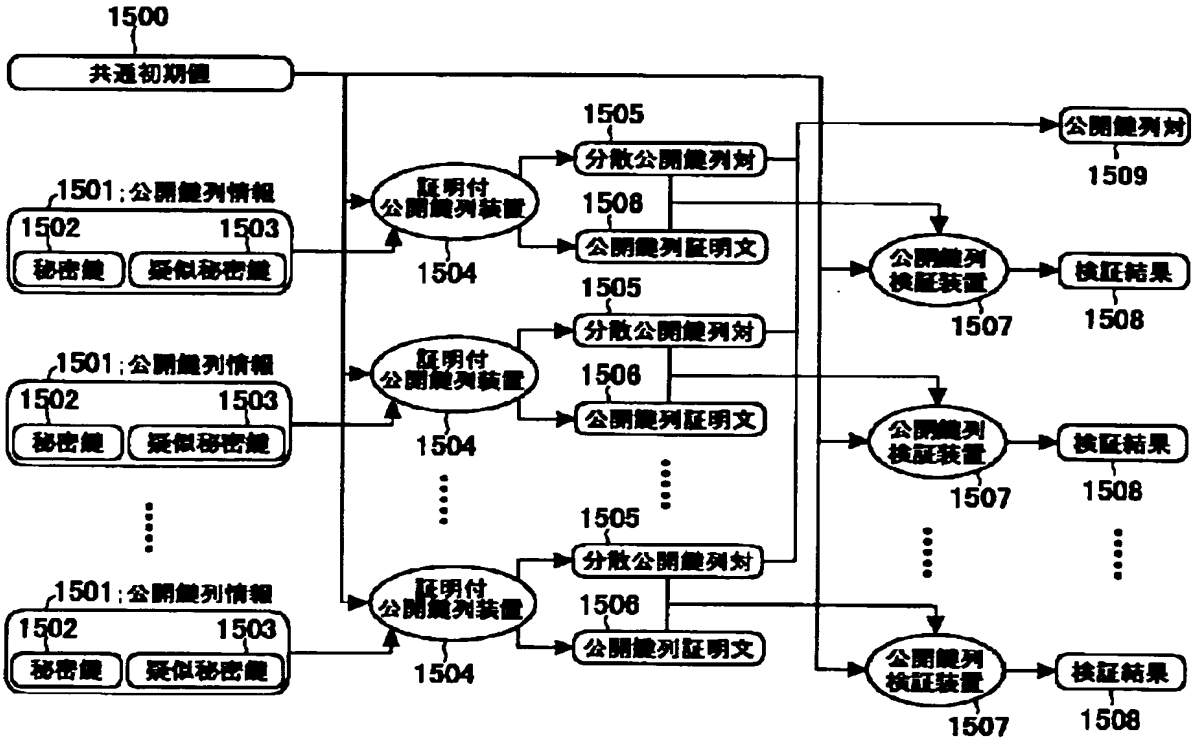
【図１３】


入力暗号文列
301
公開鍵
302
1310:共通初期値
1300:公開鍵列情報
1301
1302
秘密鍵
疑似秘密鍵
証明付
公開鍵列装置
1304
1305
分散公開鍵列対
1306
公開鍵列証明文
前処理装置
1309
入力文列
300
公開鍵列
検証装置
1307
検証結果
1308

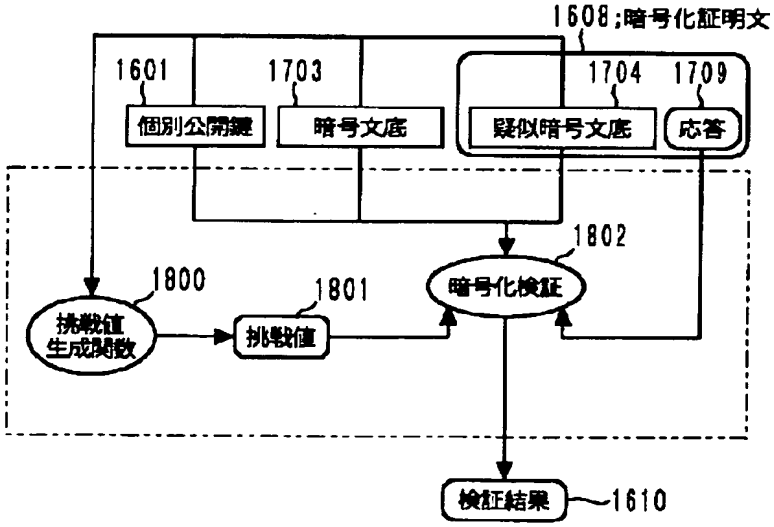
【図１４】


分散公開鍵列対
1305
基底生成関数
1400
1404:公開鍵列
公開鍵列底
公開鍵列対
1401
1403
公開鍵
302
301:入力暗号文列
入力暗号文
n個
前処理
1402
300:入力文列
322；入力暗号文列
公開鍵
323

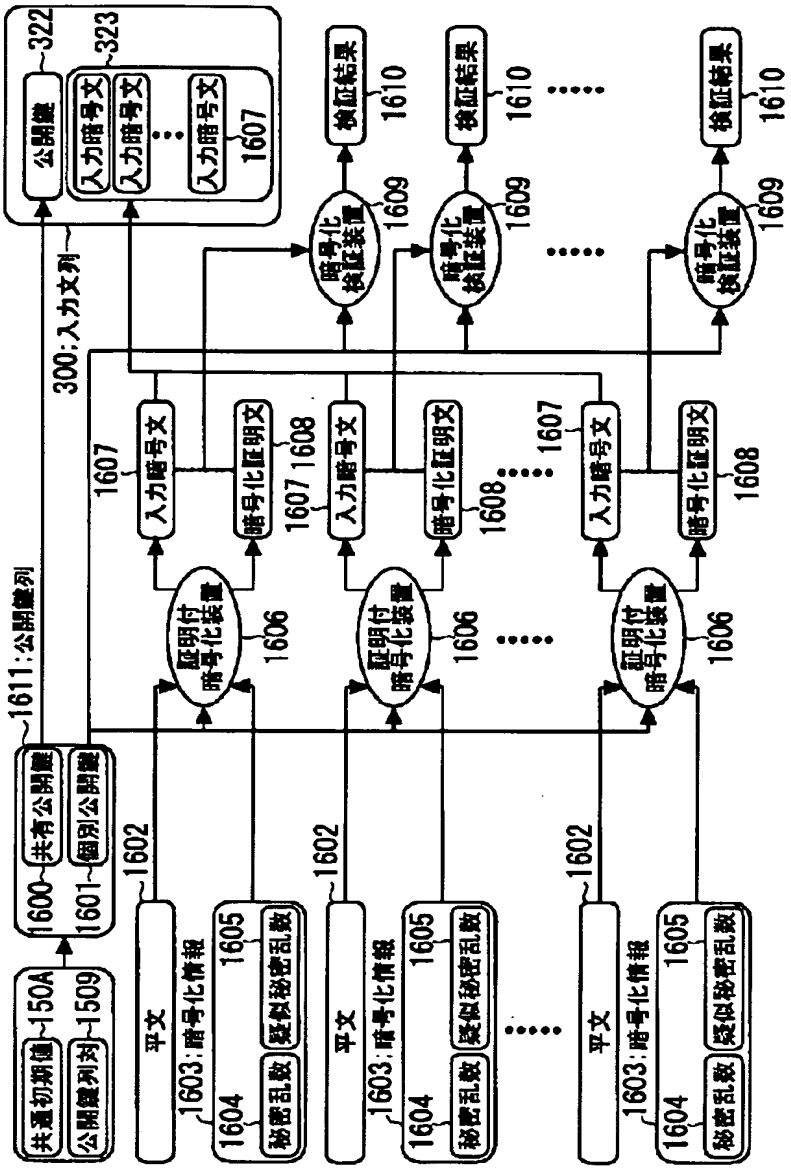
【図１５】


1500
共通初期値
1501;公開鍵列情報
1502
秘密鍵
1503
疑似秘密鍵
証明付公開鍵列装置
1504
1505
分散公開鍵列対
1508
公開鍵列証明文
1506
公開鍵列検証装置
1507
公開鍵列対
1509
検証結果
1508

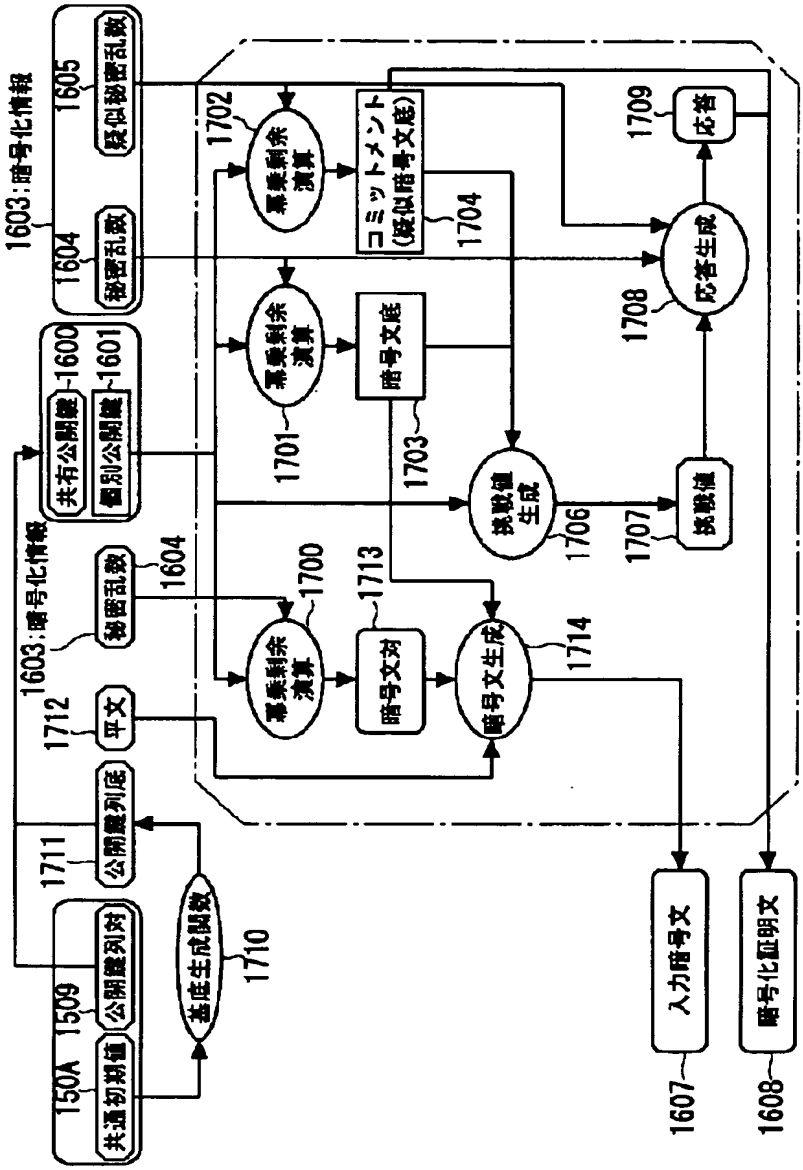
【図１８】


1608;暗号化証明文
1601
個別公開鍵
1703
暗号文底
1704
疑似暗号文底
1709
応答
1802
暗号化検証
1800
挑戦値生成関数
1801
挑戦値
検証結果
1610

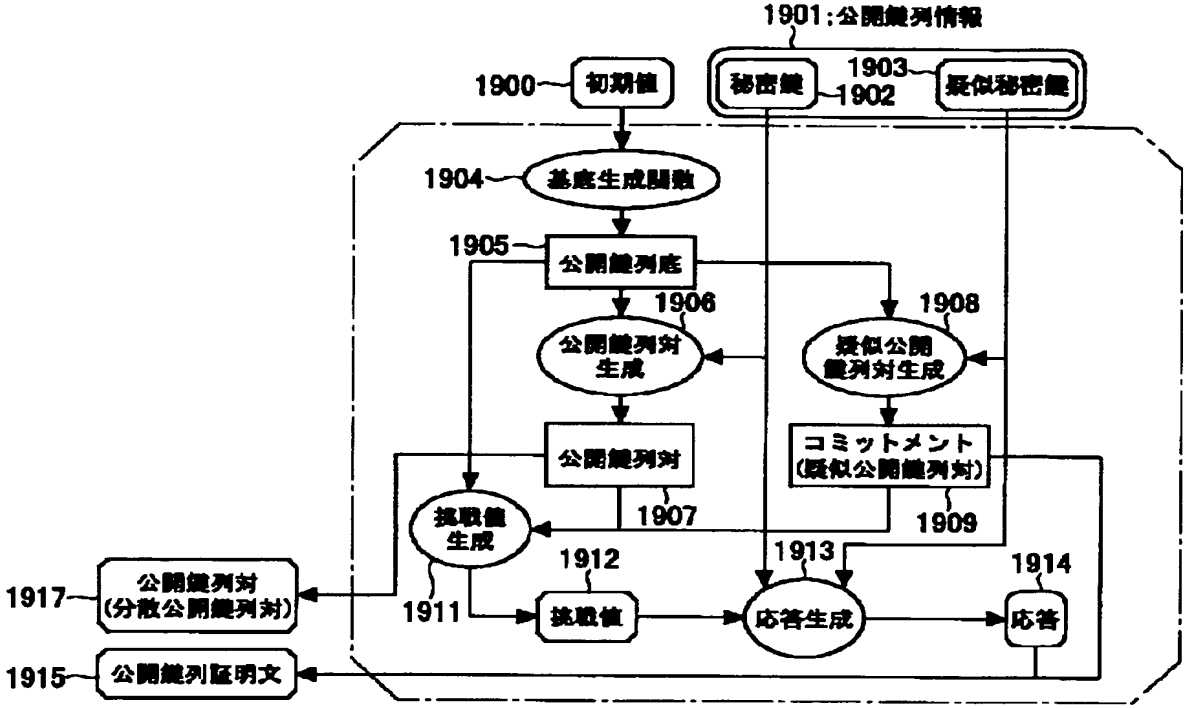
【図１６】


322
323
公開鍵
入力暗号文
入力暗号文
入力暗号文
1607
検証結果
1610
暗号化検証装置
1609
300:入力文列
1611:公開鍵列
1607
入力暗号文
暗号化証明文
1607 1608
証明付暗号化装置
1606
1602
1600
共有公開鍵
1601
個別公開鍵
150A
共通初期値
1509
公開鍵列対
平文
1603:暗号化情報
1605
1604
秘密乱数
疑似秘密乱数

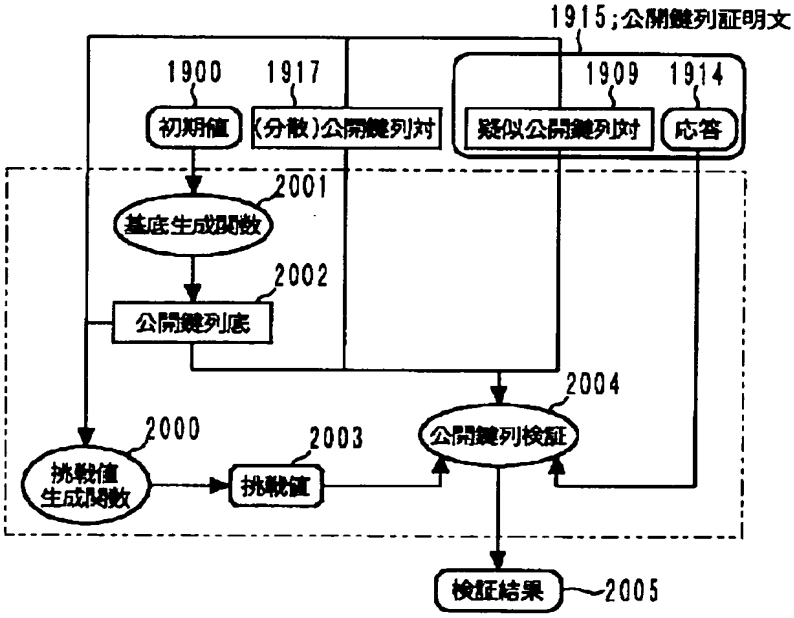
【図１７】


1603:暗号化情報
1605
疑似秘密乱数
1604
秘密乱数
1600
共有公開鍵
1601
個別公開鍵
1603:暗号化情報
1604
秘密乱数
1712
平文
1711
公開鍵列底
150A
共通初期値
1509
公開鍵列対
基底生成関数
1710
1702
冪乗剰余演算
コミットメント(疑似暗号文底)
1704
1701
冪乗剰余演算
暗号文底
1703
1700
冪乗剰余演算
1713
暗号文対
暗号文生成
1714
挑戦値生成
1706
1709
応答
1708
応答生成
1707
挑戦値
1607
入力暗号文
1608
暗号化証明文

【図１９】


1901:公開鍵列情報
1900
初期値
秘密鍵
1903
1902
疑似秘密鍵
1904
基底生成関数
1905
公開鍵列底
1906
1908
公開鍵列対生成
疑似公開鍵列対生成
公開鍵列対
コミットメント(疑似公開鍵列対)
挑戦値生成
1907
1909
1912
1913
1914
1917
公開鍵列対(分散公開鍵列対)
1911
挑戦値
応答生成
応答
1915
公開鍵列証明文

【図２０】


1915;公開鍵列証明文
1900
1917
1909
1914
初期値
(分散)公開鍵列対
疑似公開鍵列対
応答
2001
基底生成関数
2002
公開鍵列底
2004
2000
2003
公開鍵列検証
挑戦値生成関数
挑戦値
検証結果
2005

【手続補正書】
【提出日】平成１２年１１月２０日（２０００．１１．２０）
【手続補正１】
【補正対象書類名】明細書
【補正対象項目名】００３９
【補正方法】変更
【補正内容】
【００３９】例えば、入力暗号文列を、g[i, Γ]; i=1,...,n; Γ=0,...,l、公開鍵を、g[i, Γ]; i=n+1,...,n+m; Γ=0,...,l、それ以外の入力文列の成分を、g[i, Γ]; i=1,...,n+m; Γ=l+1,...,l'、一般の変換に対応する乱数(以下「情報隠蔽因子」と呼ぶ)を、A[μ, j]; μ=1,...,n+m, j=n+1,...,n+m'、再暗号化の変数を、A[i, j]; i=n+1,...,n+m, j=1,...,n、並び替えに対応する変換を表す変数を、A[i, j]; i, j=1,...,nとして、出力暗号文列　g''[i, Γ]; i=1,...,n; Γ=1,...,lを、
$g''[i,\Gamma]=\Pi_{j=1}^{n} g[j,\Gamma]^{A[j,i]} \Pi_{j=n+1}^{n+m} g[j,\Gamma]^{A[j,i]} /F^*_p$　i=1,...,n Γ=1,...,l
と生成し、変換情報保有コミットメントを、
$g''[i,\Gamma]=\Pi_{j=1}^{n} g[j,\Gamma]^{A[j,i]} \Pi_{j=n+1}^{n+m} g[j,\Gamma]^{A[j,i]} /F^*_p$　i=n+1,...,n+m'　Γ=1,...,l
と生成し、入力文列にg[i, Γ]; i=1,...,n+m; Γ=l+1,...,l'が含まれた場合の変換情報保有コミットメントを、
$g''[i,\Gamma]=\Pi_{j=1}^{n} g[j,\Gamma]^{A[j,i]} \Pi_{j=n+1}^{n+m} g[j,\Gamma]^{A[j,i]} /F^*_p$　i=1,...,n+m'　Γ=l+1,...,l'
と生成できる。
【手続補正２】
【補正対象書類名】明細書
【補正対象項目名】００４０
【補正方法】変更
【補正内容】
【００４０】また、これらをまとめて
$g''[i,\Gamma]=\Pi_{j=1}^{n+m} g[j,\Gamma]^{A[j,i]} /F^*_p$　i=1,...,n+m'　Γ=1,...,l'
と記述できる。
＊
$\Sigma_{i=1}^{n} r[i]r[i]r[i]+\rho''r'+\Sigma_{\mu=1}^{n+m} \rho'[\mu]r[\mu] /F_q$
$=\Sigma_{i=1}^{n} r[i]r[i]r[i]+\rho''(\lambda[0]+\Sigma_{i=1}^{n} \lambda[i]r[i]r[i])+\Sigma_{\mu=1}^{n+m} \rho'[\mu]r[\mu] /F_q$
$=\Sigma_{i=1}^{n} c[i]c[i]c[i]+\Sigma_{i=1}^{n} \psi[i]c[i]c[i]+\Sigma_{\mu=1}^{n+m} \phi[\mu]c[\mu] /F_q$
が成り立つことを確認し、また準応答の正当性を、検証式
$u^{r'}=u[0]\Pi_{i=1}^{n} u[i]^{r[i]r[i]} /F^*_p$
が成り立つことより確認する。
【手続補正６】
【補正対象書類名】明細書
【補正対象項目名】００７４
※
$v^{\wedge}\{\Sigma_{i=1}^{n} r[i]r[i]r[i]+\rho''r'+\Sigma_{\mu=1}^{n+m} \rho'[\mu]r[\mu]\} /F^*_p$
$=v^{\wedge}\{\Sigma_{i=1}^{n} r[i]r[i]r[i]+\rho''(\lambda[0]+\Sigma_{i=1}^{n} \lambda[i]r[i]r[i])+\Sigma_{\mu=1}^{n+m} \rho'[\mu$

＊【手続補正３】
【補正対象書類名】明細書
【補正対象項目名】００５６
【補正方法】変更
【補正内容】
【００５６】変換条件コミットメントとして、恒等式の係数または、それをコミットしたものと、準元係数をコミットしたものを生成する。実施例では恒等式の一部を、
$v, v^{\lambda[0]} /F^*_p$
のようにコミットし、準元係数を
$u, u^{\lambda[\mu]} /F^*_p$　μ=0,...,n
のようにコミットする。
【手続補正４】
【補正対象書類名】明細書
【補正対象項目名】００６８
【補正方法】変更
【補正内容】
【００６８】例えば、挑戦値c[i]; i=1,...,n+m'と応答r[i]; i=1,...,n+mが、
$\Pi_{i=1}^{n+m} 'g''[i,\Gamma]^{c[i]} =\Pi_{i=1}^{n+m} g[i,\Gamma]^{r[i]} /F^*_p$
Γ=1,...,l'
が成り立つことを確認する。
【手続補正５】
【補正対象書類名】明細書
【補正対象項目名】００７３
【補正方法】変更
【補正内容】
【００７３】例えば、変換条件コミットメントとして恒等式の係数ρ'', ρ'[μ], φ[μ], ψ[i]に対して、挑戦値c[i]; i=1,...,n+m'と、応答r[i]; i=1,...,n+mが、恒等式
$\Sigma_{i=1}^{n} r[i]r[i]+\Sigma_{\mu=1}^{n+m} \rho'[\mu]r[\mu]=\Sigma_{i=1}^{n} c[i]c[i]+\Sigma_{\mu=1}^{n+m} '\phi[\mu]c[\mu] /F_q$
や、恒等式
※【補正方法】変更
【補正内容】
【００７４】また恒等式の係数の一部がコミットされている場合は、代わりに、
$v^{\wedge}\{\Sigma_{i=1}^{n} r[i]r[i]+\Sigma_{\mu=1}^{n+m} \rho'[\mu]r[\mu]\} /F^*_p$
$=v^{\wedge}\{\Sigma_{i=1}^{n} c[i]c[i]+\Sigma_{\mu=1}^{n+m} '\phi[\mu]c[\mu]\} /F^*_p$
や、

$$]r[\mu]\}\ /F^*_p$$
$$=v\hat{}\{\Sigma_{i=1}{}^{n}c[i]c[i]c[i]+\Sigma_{i=1}{}^{n}\psi[i]c[i]c[i]+\Sigma_{\mu=1}{}^{n+m}{}'\phi[\mu]c[\mu]\}\ /F^*_p$$

が成り立つことを確認する。上式で、記号「^」は指数演算を示す。

【手続補正７】
【補正対象書類名】明細書
【補正対象項目名】００８１
【補正方法】変更
【補正内容】
【００８１】例えば、公開鍵列を、$g'[i,\Gamma];i=1,\ldots,n+m;\Gamma=1,\ldots,l$とし、入力暗号文列を、$\eta[i,\Gamma];i=1,\ldots,n;\Gamma=0,\ldots,l$、公開鍵を、$\eta[i,\Gamma];i=n+1,\ldots,n+m;\Gamma=0,\ldots,l$としたとき、入力文列　$g[i,\Gamma];i=1,\ldots,n+m;\Gamma=l+1,\ldots,l$を、検証者にも明らかな任意の正整数$s[i];i=1,\ldots,n+m$を用いて、
$g[i,\Gamma]=\eta[i,\Gamma]g'[i,\Gamma]^{s[i]}\ /F^*_p$
と表す。s[i]として、例えばs[n+m]=0,s[j]=1;j=1,...,n+m-1を選ぶ。

【手続補正８】
【補正対象書類名】明細書
【補正対象項目名】００８３
【補正方法】変更
【補正内容】
【００８３】例えば、公開鍵列を、$g'[i,\Gamma];i=1,\ldots,n+m;\Gamma=0,1$とし、平文を、$m[i];i=1,\ldots,n;\Gamma=0,1$としたとき、入力暗号文を、
$\eta[i,0]=g'[i,0]^{s[i]}\ /F^*_p,\quad i=1,\ldots,n$
$\eta[i,1]=m[i]g'[i,1]^{s[i]}\ /F^*_p,\quad i=1,\ldots,n$
と生成し、合わせて、$\eta[i,0]=g'[i,0]^{s[i]}\ /F^*_p$,　となるs[i]の知識を証明することで、g'[i,0]を使って暗号化した証明文とする。

【手続補正９】
【補正対象書類名】明細書
【補正対象項目名】０１０２
【補正方法】変更
【補正内容】
【０１０２】挑戦値生成関数の出力は、n+1個の1,0でないq以下の整数、基底生成関数の出力は、n+1個の1,0でないp以下の整数で位数qのF^*_pの元(位数p-1の乗法群の位数q部分群の元)である整数である。

【手続補正１０】
【補正対象書類名】明細書
【補正対象項目名】０１０５
【補正方法】変更
【補正内容】
【０１０５】挑戦値生成関数の場合は、|q|ビットを出力するハッシュ関数を用いて数列を求め、その中から順に1,0でないものをn+1個選んでいく(この場合、k乗する操作は必要無い)。
[公開鍵] 公開鍵について説明する。公開鍵は、二つの数値$\eta[0,0],\eta[0,1]$であり$\eta[0,0]$は位数qのF^*_pの元とする。$\eta[0,1]$は秘密鍵xを用いて、
$\eta[0,1]=\eta[0,0]^x\ /F^*_p$
と計算される。

【手続補正１１】
【補正対象書類名】明細書
【補正対象項目名】０１０６
【補正方法】変更
【補正内容】
【０１０６】[入力暗号文] 入力暗号文について説明する。平文を、p以下で位数qのF^*_pの元から選び、これをMとする。これから疑似乱数生成器で生成した秘密乱数rを用いて、入力暗号文を、
$(\eta[0,0]^r, M\ \eta[0,1]^r)\ /F^*_p$
と計算する。

【手続補正１２】
【補正対象書類名】明細書
【補正対象項目名】０１０７
【補正方法】変更
【補正内容】
【０１０７】[再暗号化] 再暗号化について説明する。Elgamal暗号文$(\eta[0,0]^r, M\ \eta[0,1]^r)\ /F^*_p$が与えられた時、任意の乱数sを選んで、
$(\eta[0,0]^r, M\eta[0,1]^r)\rightarrow(\eta[0,0]^r\ \eta[0,0]^s, M\eta[0,1]^r\ \eta[0,1]^s)\ /F^*_p=(\eta[0,0]^{r+s}, M\eta[0,1]^{r+s})\ /F^*_p$
なる変換を行うことを「再暗号化」という。上記変換はrを知らなくても実行できる。またこの変換により再暗号化された暗号文の復号結果はかわらない。この時の乱数sを、「再暗号秘密乱数」と呼ぶ。

【手続補正１３】
【補正対象書類名】明細書
【補正対象項目名】０１１０
【補正方法】変更
【補正内容】
【０１１０】[準置換行列] 準置換行列ついて説明する。「準置換行列」とは、置換行列の1である成分を、F^*_p上の３個ある１の３乗根のいずれかで置き換えたものと定義する。これらを$w, w^2, 1$として下に準置換行列の例をあげる。

【手続補正１４】
【補正対象書類名】明細書
【補正対象項目名】０１１２
【補正方法】変更
【補正内容】
【０１１２】[再暗号シャッフル] 再暗号シャッフルについて説明する。入力暗号文列$\eta[i,0],\eta[i,1];i=1,\ldots,n$の順序を入れ替えて、暗号文列$\eta'[i,0],\eta'[i,1];i=1,\ldots,n$を生成し、さらにn個の秘密乱数s[i]:i=

1,...,nと、公開鍵η[0,0], η[0,1]を用いて、出力暗号文列g''[i,Γ];i=1,...,n,Γ=0,1を、
g''[i,Γ]=η'[i,Γ]η[0,Γ]$^{s(i)}$　/F*$_p$　i=1,...,n,Γ=0,1
と計算する。これが、再暗号シャッフルの出力結果である。これを、「出力暗号文列」と呼ぶ。
【手続補正１５】
【補正対象書類名】明細書
【補正対象項目名】０１１５
【補正方法】変更
【補正内容】
【０１１５】g''[μ,Γ]=Π$_{\nu=0}^{n}$g[ν,Γ]A[ν,μ] /F*$_p$　μ=0,...,n,Γ=0,1
ここで、シャッフル行列が置換行列の場合、出力暗号文列を、g''[i,0],g''[i,1];i=1,...,nとし、展開すると、ある置換(i,j|π(i)=j)に対して、
g''[j,0]=g[i,0]η[0,0]$^{A[0,j]}$　/F*$_p$
g''[j,1]=g[i,1]η[0,1]$^{A[0,j]}$　/F*$_p$
となり、これは再暗号シャッフルの出力となる。
【手続補正１６】
【補正対象書類名】明細書
【補正対象項目名】０１１６
【補正方法】変更
【補正内容】
【０１１６】またシャッフル行列が準置換行列の場合、準再暗号シャッフルの結果
g''[j,0]=g[i,0]$^{w(i)}$　η[0,0]$^{A[0,j]}$　/F*$_p$
g''[j,1]=g[i,1]$^{w(i)}$　η[0,1]$^{A[0,j]}$　/F*$_p$
を出力する(準再暗号シャッフルとは各出力暗号文を1またはwまたはw^2乗すると再暗号シャッフルとなるものと定義する)。ここでw[i];i=1,...,nはFq上の1の三乗根のいずれかをとる。
【手続補正１７】
【補正対象書類名】明細書
【補正対象項目名】０１２２
【補正方法】変更
【補正内容】
【０１２２】係数基底404　ν、元係数403 r'[0]を生成に関しては、疑似乱数生成器で1,0でないFq上の数を生成し、r'[0]とし、疑似乱数生成器により/F*$_p$の元を生成しF*$_p$上でそのk乗をとり1,0でないものを選び位数qのF*$_p$の元を生成し、νとする。
【手続補正１８】
【補正対象書類名】明細書
【補正対象項目名】０１２３
【補正方法】変更
【補正内容】
【０１２３】r'[0]∈$_R$　Fq　,≠0,1
ν∈$_R$　F*$_p$　,≠1　,s.t.　ν^q=1 /F*$_p$
入力暗号文列η[i,0], η[i,1];i=1,...,nと、公開鍵η[0,0], η[0,1]より、入力文列400
g[μ,Γ];μ=0,...,n;Γ=0,1を、
g[0,Γ]=η[0,Γ]　　Γ=0,1
g[i,Γ]=η[i,Γ] /F*$_p$　i=1,...,n,Γ=0,1
とする。
【手続補正１９】
【補正対象書類名】明細書
【補正対象項目名】０１２８
【補正方法】変更
【補正内容】
【０１２８】さらに、係数基底404　νを用いて、隠蔽処理411により、恒等式係数410　r'[0], φ[0]を、
ν'=ν$^{r'[0]}$　/F*$_p$
ω=ν$^{\phi[0]}$　/F*$_p$
とコミットする。
【手続補正２０】
【補正対象書類名】明細書
【補正対象項目名】０１３６
【補正方法】変更
【補正内容】
【０１３６】変換条件検証処理506により、挑戦値501と応答416と変換条件コミットメント412とを用いて検証式
ν'$^{r[0]}$　ν^{Σ$_{i=1}^{n}$r[i]r[i]}=ων^{Σ$_{i=1}^{n}$(c[i]c[i]+φ[i]c[i])}　/F*$_p$
が成り立つことを確認503する。
【手続補正２１】
【補正対象書類名】明細書
【補正対象項目名】０１４６
【補正方法】変更
【補正内容】
【０１４６】ρ'∈$_R$Fq　,≠0,1
ρ''∈$_R$　Fq　,≠0,1
ν∈$_R$　F*$_p$　,≠1,　s.t.　ν^q=1 /F*$_p$
λ[μ]∈$_R$　Fq　,≠0,1,　μ=0,...,n
u∈$_R$　F*$_p$,≠1,　s.t .u^q=1 /F*$_p$
【手続補正２２】
【補正対象書類名】明細書
【補正対象項目名】０１５１
【補正方法】変更
【補正内容】
【０１５１】さらに、係数基底604　νを用いて、隠蔽処理608により、恒等式係数607　ρ', ρ'', φ[0]を、
ω=ν$^{\phi[0]}$　/F*$_p$
ν''=ν$^{\rho''}$　/F*$_p$
ν'=ν$^{\rho'}$　/F*$_p$
とコミット609する。さらに、係数基底605　uを用いて、準元係数606　λ[μ];μ=0,...,nを、
u[0]=u$^{\lambda[0]}$　/F*$_p$
u[i]=u$^{\lambda[i]}$　/F*$_p$　i=1,...,n
とコミット612する。

【手続補正２３】
【補正対象書類名】明細書
【補正対象項目名】０１５８
【補正方法】変更
【補正内容】
【０１５８】変換情報保有検証処理710により、この挑戦値705を用いて、入力文列600と、変換情報保有コミットメント605と、出力暗号文列604である出力文列603と、応答618と、を用いて検証式、
$\Pi_{\mu=0}^{n} g[\mu,\Gamma]^{r[\mu]} = \Pi_{\mu=0}^{n} g''[\mu,\Gamma]^{c[\mu]} \ /F^*_p$
$\Gamma=0,1$
が成り立つことを確認706する。
【手続補正２４】
【補正対象書類名】明細書
【補正対象項目名】０１５９
【補正方法】変更
【補正内容】
【０１５９】変換条件検証処理711により、挑戦値705と、応答618と、変換条件コミットメント609、612と、を用いて検証式、
$v''^{r'}\ v'^{r[0]}\ v\hat{}\{\Sigma_{i=1}^{n} r[i]r[i]r[i]\} = \omega v\hat{}\{\Sigma_{i=1}^{n}(c[i]c[i]c[i]+\psi[i]c[i]c[i]+\phi[i]c[i])\}\ /F^*_p$
と検証式707
$u^{r'} = \Pi_{i=1}^{n} u[i]^{r[i]r[i]} \ /F^*_p$
が成り立つこと確認708する。
【手続補正２５】
【補正対象書類名】明細書
【補正対象項目名】０１６１
【補正方法】変更
【補正内容】
【０１６１】上記証明付再暗号シャッフル方法は、入力文列に対する再暗号シャッフル行列変換が少なくとも置換行列に属するシャッフル行列を持つ再暗号シャッフル行列により行われたことを保証する効果がある。この時、出力暗号文列$g''[i\Gamma];i=1,\ldots,n;\Gamma=0,1$は、
$g''[j,0]=g[i,0]^{w[i]}\ g[0,0]^{A[0,j]} \ /F^*_p$
$g''[j,1]=g[i,1]^{w[i]}\ g[0,1]^{A[0,j]} \ /F^*_p$
を出力した可能性を排除できない。ここで、$w[i]$が全て1の時が、再暗号シャッフルである。なお、$w[i];i=1,\ldots,n$は、F_q上の1の三乗根のいずれかをとる。
【手続補正２６】
【補正対象書類名】明細書
【補正対象項目名】０１６８
【補正方法】変更
【補正内容】
【０１６８】元係数803　$r'[-1],r'[0]$、元係数805　ρ,ρ',ρ''、係数基底804　v、係数基底806　u、準元係数807　$\lambda[\mu];\mu=0,\ldots,n$に関しても、実施例(1)と同様な手法で、$r'[-1],r'[0],\rho,\rho',\rho'',\lambda[\mu];\mu=0,\ldots,n$には、1,0でない$F_q$上の数を、係数基底u,vには、位数qの$F^*_p$の元を生成する。
【手続補正２７】
【補正対象書類名】明細書
【補正対象項目名】０１６９
【補正方法】変更
【補正内容】
【０１６９】$r'[-1]\in_R F_q,\neq 0,1$
$r'[0],\in_R F_q,\neq 0,1$
$\rho\in_R F_q,\neq 0,1$
$\rho'\in_R F_q,\neq 0,1$
$\rho''\in_R F_q,\neq 0,1$
$v\in_R F^*_p,\neq 0,1,\text{s.t.}\quad v^q=1\ /F^*_p$
$\lambda[\mu]\in_R F_q,\neq 0,1\ \mu=0,\ldots,n$
$u\in_R F^*_p,\neq 0,1,\text{s.t.}\quad u^q=1\ /F^*_p$
【手続補正２８】
【補正対象書類名】明細書
【補正対象項目名】０１７２
【補正方法】変更
【補正内容】
【０１７２】変換情報保有コミットメント生成処理832における再暗号シャッフル行列作用808により、上記再暗号シャッフル行列802を入力文列800に以下の様に作用させて、出力文列809$g''[\mu,\Gamma];\mu=0,\ldots,n;\Gamma=0,1$を、
$g''[\mu,\Gamma]=\Pi_{\nu=1}^{n} g[\nu,\Gamma]^{A[\nu,\mu]} \ /F^*_p \quad \mu=0,\ldots,n,\Gamma=0,1$
と生成する。ここで、$g''[i,\Gamma];i=1,\ldots,n;\Gamma=0,1$を、出力暗号文列810、$g''[0,\Gamma];\Gamma=0,1$を、変換情報保有コミットメント811とする。
【手続補正２９】
【補正対象書類名】明細書
【補正対象項目名】０１７５
【補正方法】変更
【補正内容】
【０１７５】さらに係数基底804　vを用いて、隠蔽処理814、818により、恒等式係数813,817　$r'[-1],r'[0],\underline{\phi}[0],\phi[0],\rho,\rho',\rho''$を、
$\omega=v^{\phi[0]} \ /F^*_p$
$v''=v^{\rho''} \ /F^*_p$
$v'=v^{\rho'} \ /F^*_p$
$\omega'=v^{\rho} \ /F^*_p$
とコミット819し、
$\underline{v}=v^{r'[-1]} \ /F^*_p$
$\underline{v'}=v^{r'[0]} \ /F^*_p$
$\underline{\omega}=v^{\underline{\phi}[0]} \ /F^*_p$
とコミット815する。
【手続補正３０】
【補正対象書類名】明細書
【補正対象項目名】０１７６
【補正方法】変更

【補正内容】
【０１７６】さらに、係数基底806 uを用いて、準元係数807 $\lambda[\mu];\mu=0,\ldots,n$を、
$u[0]=u^{\lambda^{[0]}}/F^*_p$
$u[i]=u^{\lambda^{[i]}}/F^*_p \quad i=1,\ldots,n$
とコミット821、820する。
【手続補正３１】
【補正対象書類名】明細書
【補正対象項目名】０１７９
【補正方法】変更
【補正内容】
【０１７９】応答生成処理835により、以上の入力文列800と、出力暗号文列810と、コミットメント823を、挑戦値生成関数824の引数として、挑戦値825を、
$c[0]=1$
$c[i]=Hash[i](g[\mu,\Gamma],g''[\nu,\Gamma],u[\nu],u,\phi[j],\psi[j],\omega,\omega',v',v'',v,\underline{\phi}[j],\underline{\omega},\underline{v'},\underline{v};\mu=-1,\ldots,n;\nu=0,\ldots,n;j=1,\ldots,n;\Gamma=0,1,2)i=1,\ldots,n$
と生成し、この挑戦値825から、再暗号シャッフル行列802を用いて、応答827を、
$r[\mu]=\Sigma_{\nu=0}^{n}A[\mu,\nu]c[\nu]/F_q \quad \mu=-1,\ldots,n$
と生成826する。
【手続補正３２】
【補正対象書類名】明細書
【補正対象項目名】０１８４
【補正方法】変更
【補正内容】
【０１８４】変換情報保有検証処理907により、この挑戦値901を用いて入力文列800と、変換情報保有コミットメント811と、出力暗号文列810である出力文列809と、応答827とを用いて検証式、
$\Pi_{\mu=-1}^{n}g[\mu,\Gamma]^{r[\mu]}=\Pi_{\mu=0}^{n}g''[\mu,\Gamma]^{c[\mu]}/F^*_p \quad \Gamma=0,1$
が成り立つことを確認902する。
【手続補正３３】
【補正対象書類名】明細書
【補正対象項目名】０１８５
【補正方法】変更
【補正内容】
【０１８５】変換条件検証処理908、909により、挑戦値901と応答827と準応答829と変換条件コミットメント815、819、821とを用いて、検証式
$v'^{r'}v'^{r[0]}\omega'^{r[-1]}\hat{v}\{\Sigma_{i=1}^{n}r[i]r[i]r[i]\}=\omega\hat{v}\{\Sigma_{i=1}^{n}(c[i]c[i]c[i]+\psi[i]c[i]c[i]+\phi[i]c[i])\}/F^*_p$
が成り立つことを確認904し、検証式
$u^{r'}=u[0]\Pi_{i=1}^{n}u[i]^{r[i]r[i]}/F^*_p$
が成り立つことを確認905し、検証式
$\underline{v'}^{r[0]}\underline{v}^{r[-1]}\hat{v}\{\Sigma_{i=1}^{n}r[i]r[i]\}=\underline{\omega}\hat{v}\{\Sigma_{i=1}^{n}(c[i]c[i]+\underline{\phi}[i]c[i])\}/F^*_p$
が成り立つこと確認903する。
【手続補正３４】
【補正対象書類名】明細書
【補正対象項目名】０１９４
【補正方法】変更
【補正内容】
【０１９４】$\rho'\in_R F_q,\neq 0,1$
$\rho''\in_R F_q,\neq 0,1$
$r'[0],\in_R F_q,\neq 0,1$
$v\in_R F^*_p,\neq 1,\text{s.t.}\ v^q=1/F^*_p$
$\lambda[\mu]\in_R F_q,\neq 0,1 \quad \mu=0,\ldots,n$
$u\in_R F^*_p,\neq 1,\text{s.t.}\ u^q=1/F^*_p$
【手続補正３５】
【補正対象書類名】明細書
【補正対象項目名】０１９７
【補正方法】変更
【補正内容】
【０１９７】変換情報保有コミットメント生成処理1042における再暗号シャッフル行列作用1009により、上記暗号シャッフル行列1001を入力文列1000に以下の様に作用させて、出力文列1010 $g''[\mu,\Gamma];\mu=0,\ldots,n;\Gamma=0,1$を、
$g''[\mu,\Gamma]=\Pi_{\nu=0}^{n}g[\nu,\Gamma]^{A[\nu,\mu]}/F^*_p \quad \mu=0,\ldots,n,\Gamma=0,1$
と生成する。
【手続補正３６】
【補正対象書類名】明細書
【補正対象項目名】０２００
【補正方法】変更
【補正内容】
【０２００】第２の変換情報保有コミットメント1021 $\underline{g''}[0,\Gamma']$を、
$\underline{g''}[0,\Gamma']=\Pi_{\nu=0}^{n}g[\nu,\Gamma']^{A[\nu,0]}/F^*_p \quad \Gamma=0\text{or}1$
と生成1020する。
【手続補正３７】
【補正対象書類名】明細書
【補正対象項目名】０２０２
【補正方法】変更
【補正内容】
【０２０２】さらに、係数基底1003vを用いて、隠蔽処理1024により、恒等式係数1023$\phi[0],\rho',\rho''$を、
$\omega=v^{\phi^{[0]}}/F^*_p$
$v''=v^{\rho''}/F^*_p$
$v'=v^{\rho'}/F^*_p$
とコミット1025する。
【手続補正３８】
【補正対象書類名】明細書
【補正対象項目名】０２０３
【補正方法】変更
【補正内容】

【０２０３】さらに、係数基底1008uを用いて、準元係数1007λ[μ]；μ=0,...,nを、
u[0]=u^{λ[0]}　/F*p
u[i]=u^{λ[i]}　/F*p　　i=1,...,n
とコミット1027する。
【手続補正３９】
【補正対象書類名】明細書
【補正対象項目名】０２０５
【補正方法】変更
【補正内容】
【０２０５】さらに係数基底1003　vを用いて、隠蔽処理1015により、恒等式係数1014　r'[0],φ[0]を、
v'=v^{r'[0]}　/F*p
ω=v^{φ[0]}　/F*p
とコミット1016する。
【手続補正４０】
【補正対象書類名】明細書
【補正対象項目名】０２１５
【補正方法】変更
【補正内容】
【０２１５】変換情報保有検証処理1112により、第１の挑戦値1101を用いて、入力文列1000と、第１の変換情報保有コミットメント1012と、出力暗号文列1011と、第１の応答1033と、を用いて検証式、
Π_{μ=0}^{n} g[μ,Γ]^{r[μ]}=Π_{μ=0}^{n} g''[μ,Γ]^{c[μ]}　/F*p
Γ=0,1
が成り立つことを確認1103する。
【手続補正４１】
【補正対象書類名】明細書
【補正対象項目名】０２１６
【補正方法】変更
【補正内容】
【０２１６】変換情報保有検証処理1113により、第２の挑戦値1109を用いて第２の入力文列1019と、第２の変換情報保有コミットメント1021と、出力暗号文列1011と、第２の応答1037と、を用いて第２の知識検証式、
Π_{μ=0}^{n} g[μ,Γ']^{r[μ]}=g''[0,Γ']Π_{i=1}^{n} g''[i,Γ']^{c[i]}　/F*p　　Γ'=0
が成り立つことを確認1105する。
【手続補正４２】
【補正対象書類名】明細書
【補正対象項目名】０２１７
【補正方法】変更
【補正内容】
【０２１７】変換条件検証処理1111により、第1の挑戦値1101と、第1の応答1033と、第1の変換条件コミットメント1025とを用いて、検証式1102、
v''^{r'} v'^{r[0]} v^{Σ_{i=1}^{n} r[i]r[i]r[i]}=ω v^{Σ_{i=1}^{n}(c[i]c[i]c[i]+ψ[i]c[i]c[i]+φ[i]c[i])}　/F*p
と、準応答1039と、準応答コミットメント1027と、第1の応答1033と、検証式1107、
u^{r'}=u[0]Π_{i=1}^{n} u[i]^{r[i]r[i]}　/F*p
が成り立つことを確認する。
【手続補正４３】
【補正対象書類名】明細書
【補正対象項目名】０２１８
【補正方法】変更
【補正内容】
【０２１８】変換条件検証処理1114により、第２の挑戦値1109と、第２の応答1037と、第２の変換条件コミットメント1016とを用いて、検証式1106、
v'^{r[0]} v^{Σ_{i=1}^{n} r[i]r[i]}=ω v^{Σ_{i=1}^{n}(c[i]c[i]+φ[i]c[i])}　/F*p
が成り立つことを確認する。
【手続補正４４】
【補正対象書類名】明細書
【補正対象項目名】０２２５
【補正方法】変更
【補正内容】
【０２２５】入力暗号文列301　η[i,0]、η[i,1]；i=1,...,nと、公開鍵302　η[0,0]、η[0,1]が入力され、公開鍵302およびElgamal領域変数p、qより、基底生成関数1200で、入力ベクトル1201を生成し、入力文列300　g[μ,Γ]；μ=0,...,n；Γ=0,1,2を、
g[0,Γ]=η[0,Γ]　　Γ=0,1
g[i,Γ]=η[i,Γ]/F*p　　i=1,...,n,　Γ=0,1
g[μ,2]=Hash'[μ](p,q,η[0,0],g[0,1,Λ]；Λ=1,...,t)μ=0,...,n
とする。
【手続補正４５】
【補正対象書類名】明細書
【補正対象項目名】０２２６
【補正方法】変更
【補正内容】
【０２２６】前記実施例(1)から実施例(4)に、本実施例の入力文列生成方法を適用する場合、これらの実施例におけるΓの値をとる範囲を、全て0、1から、0、1、2に変更する。この新たに導入された、入力暗号文でも公開鍵でもないΓ＝２の成分が、入力暗号文生成者にも意図できない入力文列の成分となり、証明者が生成できる応答に制限を課す働きをし、入力暗号文生成者と再暗号シャッフル証明文生成者とが共謀して再暗号シャッフル証明文の偽造を行うことを阻止する。
【手続補正４６】
【補正対象書類名】明細書
【補正対象項目名】０２２７
【補正方法】変更
【補正内容】
【０２２７】また前記実施例(3)に、本実施例の入力文列生成方法を適用する場合、入力文列を、g[-1,Γ]まで

拡張して、公開鍵　g[-1,0]、g[-11]、η[0,0]、η[0,1]より、
g[-1,Γ]=η[-1,Γ]　　Γ=0,1
g[0,Γ]=η[0,Γ]　　Γ=0,1
g[i,Γ]=η[i,Γ]/F*p,　　i=1,...,n,Γ=0,1
g[μ,2]=Hash'[μ](p,q,η[0,0],g[0,1,Λ];Λ=1,...,t)　μ=-1,...,n
とする。
【手続補正４７】
【補正対象書類名】明細書
【補正対象項目名】０２２８
【補正方法】変更
【補正内容】
【０２２８】また実施例(4)に、本実施例の入力文列生成方法を適用する場合、Γ'=２とし、第２の情報隠蔽因子より、第２の変換情報保有コミットメントを、
g''[02]=Πν=1^n g[ν,2]^A'[ν,0] /F*p,
g''[i2]=Πν=1^n g[ν,2]^A[ν,i] /F*p,　　i=1,...,n
と変更する。
【手続補正４８】
【補正対象書類名】明細書
【補正対象項目名】０２３０
【補正方法】変更
【補正内容】
【０２３０】さらに、変換情報保有検証処理における第２の知識検証式は、
Πμ=0^n g[μ,2]^r[μ]=Πμ=0^n g''[μ,Γ']^c[μ] /F*p,
と変更する。
【手続補正４９】
【補正対象書類名】明細書
【補正対象項目名】０２３３
【補正方法】変更
【補正内容】
【０２３３】公開鍵列検証方法1307により、各証明者の出力した分散公開鍵列対1305と、公開鍵列証明文と、共通初期値1310とから、分散公開鍵列1305の正当性が検証されたら、前処理方法により、各証明者の分散公開鍵列対1305　g'[μ,1,Λ];μ=0,...,n;Λ=1,...,tを合わせて、公開鍵列対140　3g'[μ,1,Λ];μ=0,...,nを、
g'[μ,1]=ΠΛ=1^t g'[μ,1,Λ] /F*p,　　μ=0,...,n
とする。ここで、g'[0,1]=η[0,1]に入れ替える。
【手続補正５０】
【補正対象書類名】明細書
【補正対象項目名】０２３６
【補正方法】変更
【補正内容】
【０２３６】公開鍵列1404と、入力暗号文列301と、公開鍵302とから、入力文列300　g[μ,Γ];μ=0,...,n;Γ=0,1を、
g[0,Γ]=η[0,Γ]　　Γ=0,1
g[i,Γ]=η[i,Γ]g'[i,Γ]/F*p,　　i=1,...,n,Γ=0,1
とする（前処理1402）。
【手続補正５１】
【補正対象書類名】明細書
【補正対象項目名】０２３７
【補正方法】変更
【補正内容】
【０２３７】前記実施例(3)に、本実施例の入力文列生成方法を適用する場合、入力暗号文列　η[i,Γ];i=1,...,n;Γ=0,1と、公開鍵　η[0,Γ];Γ=0,1に対して、公開鍵列　g'[μ,Γ];μ=-1,...,n;Γ=0,1を生成する。ただし、g'[0,Γ];Γ=0,1は、公開鍵に等しい。そして、入力文列　g[μ,Γ];μ=-1,...,n;Γ=0,1を、
g[-1,Γ]=η[0,Γ]　　Γ=0,1
g[i,Γ]=η[i,Γ]g'[i,Γ]/F*p,　　i=0,...,n,Γ=0,1
とする。
【手続補正５２】
【補正対象書類名】明細書
【補正対象項目名】０２４２
【補正方法】変更
【補正内容】
【０２４２】公開鍵列検証方法1507により、各証明者の出力した分散公開鍵列対1505と、公開鍵列証明文1506と、共通初期値1500より、分散公開鍵列対1505の正当性が検証1508されたら、各証明者の分散公開鍵列対150　5g'[μ,1,Λ];μ=0,...,n;Λ=1,...,tを合わせて、公開鍵列対1509　g'[μ,1,Λ];μ=0,...,nを、
g'[μ,1]=ΠΛ=1^t g'[μ,1,Λ] /F*p,　　μ=0,...,n
とする。
【手続補正５３】
【補正対象書類名】明細書
【補正対象項目名】０２４５
【補正方法】変更
【補正内容】
【０２４５】各入力暗号文生成者i=1....,nは、証明付暗号化方法1606により、平文1602m[i]と、個別公開鍵1601g'[i,Γ];Γ=0,1と、秘密乱数1604s[i]と、疑似秘密乱数1605s'[i]とより、入力暗号文1607η[i,Γ];Γ=0,1を、
η[i,0]=g'[i,0]^s[i] /F*p,
η[i,1]=m[i]g'[i,1]^s[i] /F*p,
と生成する。
【手続補正５４】
【補正対象書類名】明細書
【補正対象項目名】０２４６
【補正方法】変更
【補正内容】
【０２４６】またコミットメント(疑似暗号文底1704)、挑戦値1707、応答1709を順に以下のように生成し、
η[i2]=g'[i,0]^s'[i] /F*p,

c'[i]=Hash[0](η[i,0], η[i,1], η[i2])
θ'[i]=c'[i]s[i]+s'[i] /F_q
疑似暗号文底1704と応答1709とを暗号化証明文1608とする。
【手続補正５５】
【補正対象書類名】明細書
【補正対象項目名】０２４７
【補正方法】変更
【補正内容】
【０２４７】暗号化検証装置により、全ての入力暗号文1607と暗号化証明文1608に関して、
c'[i]=Hash[0](η[i,0], η[i,1], η[i2])
と、挑戦値1801を求め、これに応答1709を用いて、検証式1802
$\eta[i,0]^{\theta'[i]}=\eta[i,1]^{c'[i]}\ \eta[i2]\ /F^*_p$
が成り立つことを確認1610する。入力暗号文1607全ての正当性が確認されたら、入力暗号文323　η[i,Γ];Γ=0,1と、共有公開鍵1600　g'[0,Γ];Γ=0,1とより入力文列300を、
g[0,Γ]=g'[0,Γ]
g[i,Γ]=η[i,Γ]　i=1,...,n
とする。
【手続補正５６】
【補正対象書類名】明細書
【補正対象項目名】０２５４
【補正方法】変更
【補正内容】
【０２５４】これから、秘密鍵1902　xと、疑似秘密鍵1903αとにより、(分散)公開鍵列対1907　g'[μ,1];μ=0,...,nが、
$g'[\mu,1]=g'[\mu,0]^{x}\ /F^*_p\quad \mu=0,\ldots,n$
と生成1906され、疑似公開鍵列対1909が、
$g'[\mu,2]=g'[\mu,0]^{\alpha}\ /F^*_p\quad \mu=0,\ldots,n$
と生成1908される。
【手続補正５７】
【補正対象書類名】明細書
【補正対象項目名】０２５６
【補正方法】変更
【補正内容】
【０２５６】公開鍵列検証方法により、挑戦値2003を、
c''=Hash[0](g'[μ,0],g'[μ,2];μ=0,...,n)
と生成2000し、応答1914を用いて検証式、
$g'[\mu,0]^{\theta}=g'[\mu,0]^{c''}\ g'[\mu,2]\ /F^*_p\quad \mu=0,\ldots,n$
を検証2004する。
【手続補正５８】
【補正対象書類名】明細書
【補正対象項目名】０２６３
【補正方法】変更
【補正内容】
＊【０２６３】β[Λ]∈R F_q, ≠0,1
【手続補正５９】
【補正対象書類名】明細書
【補正対象項目名】０２６４
【補正方法】変更
【補正内容】
【０２６４】また、自身の公開鍵　g[0,0]、g'[0,1,Λ]を、g[0,0]、g[0,1]と、入力された暗号文列を、g[i,Γ];i=1,...,n,　Γ=0,1とし、自身の公開鍵と秘密鍵x[Λ]から、部分復号基底G[μ,0Λ];μ=0,...,nと、疑似部分復号基底G[μ,1,Λ];μ=0,...,nを、
$G[\mu,0,\Lambda]=g[\mu,0]^{x[\Lambda]}\ /F^*_p\quad \mu=0,\ldots,n$
$G[\mu,1,\Lambda]=g[\mu,0]^{\beta[\Lambda]}\ /F^*_p\quad \mu=0,\ldots,n$
と生成する。コミットメントとして、g[μ,ΓΛ];μ=0,...,n,Γ=0,1,Λ=0,...,tを出力する。
【手続補正６０】
【補正対象書類名】明細書
【補正対象項目名】０２６７
【補正方法】変更
【補正内容】
【０２６７】部分復号を
g[i,0]→g[i,0]　i=1,...,n
g[i,1]→g[i,1]/G[i,0,Λ]　/F*p,　i=1,...,n
として出力する。
【手続補正６１】
【補正対象書類名】明細書
【補正対象項目名】０２６８
【補正方法】変更
【補正内容】
【０２６８】検証処理は、入力暗号文列と証明文中とより挑戦値を、
c[Λ]=Hash[0](g[μ,0],G[μ,Γ,Λ];μ=0,...,n;Γ=0,1)
と生成し、証明文中の応答、入力暗号文列、分復号基底、疑似部分復号基底を用いて、
$g[\mu,0]^{r[\Lambda]}=G[\mu,0,\Lambda]^{c[\Lambda]}G[\mu,1,\Lambda]\ /F^*_p\quad \mu=0,\ldots,n$
を確認し、さらに部分復号が、このG[μ,0,Λ]を用いて行われたことを確認して受理する。
【手続補正６２】
【補正対象書類名】明細書
【補正対象項目名】０２７１
【補正方法】変更
【補正内容】
【０２７１】［完全性］入力文列と、出力暗号文列と変換情報保有コミットメントとである出力文列と、これに付随する応答と挑戦値が変換情報保有検証処理の検証式を満たすことは、

＊
$$\Pi_{\mu=1}^{n+s} g[\mu,\Gamma]^{r[\mu]}=\Pi_{\mu=1}^{n+s} g[\mu,\Gamma]\hat{}\{\Sigma_{\nu=1}^{n+s'} A[\mu,\nu]c[\nu]\}$$

$/F^*_p$
$=\Pi_{\nu=1}^{n+m'}(\Pi_{\mu=1}^{n+m} g[\mu,\Gamma]^{A[\mu,\nu]})^{c[\nu]} /F^*_p$
$=\Pi_{\nu=1}^{n+m'} g''[\nu,\Gamma]^{c[\nu]} /F^*_p$

より分かる。

【手続補正６３】
【補正対象書類名】明細書
【補正対象項目名】０２７２
【補正方法】変更
【補正内容】
【０２７２】準元係数をコミットしたものと、これに付随する応答と準応答が検証式を満たすことは、

$u^{r'}=u\hat{}\{\lambda[0]+\Sigma_{i=1}^{n}\lambda[i]r[i]r[i]\} /F^*_p$ ＊
$v'^{r[0]}\Pi_{i=1}^{n}v^{r[i]r[i]} /F^*_p$
$=(v^{r'[0]})^{r[0]}v\hat{}\{\Sigma_{i=1}^{n}\Sigma_{\mu=0}^{n}\Sigma_{\nu=0}^{n}A[i,\mu]A[i,\nu]c[\mu]c[\nu]\} /F^*_p$
$=v\hat{}\{r'[0]\Sigma_{\mu=0}^{n}[0,\mu]c[\mu]+2\Sigma_{i=1}^{n}\Sigma_{j=1}^{n}A[i,0]A[i,j]c[j]+\Sigma_{i=1}^{n}A[i,0]A[i,0]+\Sigma_{i=1}^{n}\Sigma_{j=1}^{n}\Sigma_{k=1}^{n}A[i,j]A[i,k]c[j]c[k]\} /F^*_p$
$=v\hat{}\{\Sigma_{i=1}^{n}\phi[i]c[i]+\phi[0]+\Sigma_{i=1}^{n}c[i]c[i]\} /F^*_p$
$=\omega v\hat{}\{\Sigma_{i=1}^{n}(c[i]c[i]+\phi[i]c[i])\} /F^*_p$

より成立することがわかる。以上、A[i,j]が置換行列であるという事実を使った。

【手続補正６５】
【補正対象書類名】明細書
【補正対象項目名】０２７５ ※
$\Sigma_{i=1}^{n}r[i]r[i]r[i]+\Sigma_{i=1}^{n}\rho''\lambda[i]r[i]r[i]+\rho'r[0] /F^*_p$
$=\Sigma_{h=1}^{n}\Sigma_{i=1}^{n}\Sigma_{j=1}^{n}\Sigma_{k=1}^{n}A[h,i]A[h,j]A[h,k]c[i]c[j]c[k]$
$+\Sigma_{h=1}^{n}\Sigma_{i=1}^{n}\Sigma_{j=1}^{n}(3A[h,0]A[h,i]A[h,j]+\rho''\lambda[h]A[h,i]A[h,j])c[i]c[j]$
$+\Sigma_{h=1}^{n}\Sigma_{i=1}^{n}(3A[h,0]A[h,0]A[h,i]+2\rho''\lambda[h]A[h,0]A[h,i]+\rho'A[0,i])c[i]$
$+\Sigma_{h=1}^{n}(A[h,0]A[h,0]A[h,0]+\rho''\lambda[h]A[h,0]A[h,0])+\rho''\lambda[0]+\rho'A[0,0]$
$/F_q$
$=\Sigma_{h=1}^{n}(c[h]c[h]c[h]+\psi[h]c[h]c[h]+\phi[i]c[i]+\phi[0]) /F_q$

であり、これは、$v\hat{}\{\Sigma_{h=1}^{n}(c[h]c[h]c[h]+\psi[h]c[h]c[h]+\phi[i]c[i])\}\omega[0] /F_p$の指数部と等しい。

【手続補正６６】
【補正対象書類名】明細書
【補正対象項目名】０２７８
【補正方法】変更
【補正内容】
【０２７８】前記実施例(8)の証明付公開鍵列方法が出力した公開鍵列底と、公開鍵列対と疑似公開鍵列対と、これに付随する応答と挑戦値が検証処理の検証式を満たすことは、

$g'[\mu,0]^{r}=g'[\mu,0]^{cx+a} /F^*_p$
$=g'[\mu,0]^{xc}g'[\mu,0]^{a} /F^*_p$
$=g'[\mu,1]^{c}g'[\mu,2] /F^*_p$

から分かる。

【手続補正６７】
【補正対象書類名】明細書
【補正対象項目名】０２８５
【補正方法】変更
【補正内容】

＊$=u^{\lambda[0]}\Pi_{i=1}^{n}(u^{\lambda[i]})^{r[i]r[i]} /F^*_p$
$=u[0]\Pi_{i=1}^{n}u[i]^{r[i]r[i]} /F^*_p$

よりわかる。

【手続補正６４】
【補正対象書類名】明細書
【補正対象項目名】０２７４
【補正方法】変更
【補正内容】
【０２７４】実施例(1)の恒等式係数に関しては、

※【補正方法】変更
【補正内容】
【０２７５】前記実施例(2)の恒等式係数に関しては、
$v''^{r'}v'^{r[0]}\Pi_{i=1}^{n}v^{r[i]r[i]r[i]} /F^*_p$
のvに対する指数部が、

【０２８５】以上により、検証式を満たすr[μ];μ=1,...,n+mとして、$g''[\nu,\Gamma]=\Pi_{\mu=1}^{n+m}g[\mu,\Gamma]^{A[\mu,\nu]} /F^*_p$;ν=1,...,n+m'なるA[μ,ν]を用いて、$r[\mu]=\Sigma_{\nu=1}^{n+m}A'[\mu,\nu]c[\nu] /F_q$;μ=1,...,n+m、と生成する以外には、証明者は計算できない。個別公開鍵を用いる方法でも同様である。

【手続補正６８】
【補正対象書類名】明細書
【補正対象項目名】０２８６
【補正方法】変更
【補正内容】
【０２８６】上述のようにあるΓに関して
$g''[\nu,\Gamma]=\Pi_{\mu=1}^{n+m}g[\mu,\Gamma]^{A[\mu,\nu]} /F^*_p$ ν=1,...,n+m'
の関係が証明されたならば、他のΓに関しても以下のように同様に証明される。

【手続補正６９】
【補正対象書類名】明細書
【補正対象項目名】０２８７
【補正方法】変更

【補正内容】
【０２８７】挑戦値生成関数の引数に含めたg[μ，Γ]，g''[ν，Γ]に対して検証式が成り立つならば、
$g''[\nu,\Gamma]=\Pi_{\mu=1}^{n+m'} g[\mu,\Gamma]^{A[\mu,\nu]} /F^*_p \quad \nu=1,\ldots,n+m'$
である。
【手続補正７０】
【補正対象書類名】明細書
【補正対象項目名】０２８８
【補正方法】変更
【補正内容】
【０２８８】なぜならば、
$g''[\nu,\Gamma]=\Pi_{\mu=1}^{n+m'} g[\mu,\Gamma]^{A'[\mu\nu]} /F^*_p \quad \nu=1,\ldots,n+m'$ ＊
＊と表した時に検証式が成り立つならば、
$=\Pi_{\mu=1}^{n+m'} g[\mu,\Gamma]^{\wedge}\{\Sigma_{\nu=1}^{n+m'} ' (A[\mu,\nu]-A'[\mu,\nu])c[\nu]\}=1 \quad /F^*_p$
が成り立つ。
【手続補正７１】
【補正対象書類名】明細書
【補正対象項目名】０２８９
【補正方法】変更
【補正内容】
【０２８９】ところが、無作為に選ばれたc[ν]に関して、これが成り立つのは、
$\Pi_{\mu=1}^{n+m'} g[\mu,\Gamma]^{A[\mu,\nu]}=\Pi_{\mu=1}^{n+m'} g[\mu,\Gamma]^{A'[\mu,\nu]} /F^*_p \quad \nu=1,\ldots,n+m'$
の時だけであるからである。

【手続補正書】
【提出日】平成１２年１２月２８日（２０００．１２．２８）
【手続補正１】
【補正対象書類名】明細書
【補正対象項目名】全文
【補正方法】変更
【補正内容】
【書類名】　　明細書
【発明の名称】　　証明付再暗号シャッフル方法と装置、再暗号シャッフル検証方法と装置、入力文列生成方法と装置及び記録媒体
【特許請求の範囲】
【請求項１】複数の暗号文と一つまたは複数の公開鍵とからなる入力文列と、再暗号シャッフル情報とを入力し、前記暗号文に対して順番の並び替えと前記公開鍵による再暗号化とを施した出力暗号文列と、上記処理に関する証明文である再暗号シャッフル証明文とを出力する証明付再暗号シャッフル方法において、
前記入力文列から出力暗号文列を生成するとともに、前記入力文列から前記出力暗号文への変換情報の保有に関するコミットメント（「変換情報保有コミットメント」という）を生成する変換情報保有コミットメント生成ステップと、
前記変換の満たす条件に関するコミットメント（「変換条件コミットメント」という）を生成する、変換条件コミットメント生成ステップと、
再暗号シャッフル情報と挑戦値とから応答を生成する、応答生成ステップと、
を含み、
前記変換情報保有コミットメントと前記変換条件コミットメントと前記応答とを前記再暗号シャッフル証明文として出力し、
前記再暗号シャッフル情報は、入力暗号文の並び替え方と、再暗号化に用いた変数と、乱数とを含む、ことを特徴とする証明付再暗号シャッフル方法。
【請求項２】入力文列と、出力暗号文列と、再暗号シャッフル証明文が入力され、受理または不受理である検証結果を出力する再暗号シャッフル検証方法において、
前記入力文列と、前記出力暗号文列と、前記入力文列から前記出力暗号文への変換情報の保有に関する変換情報保有コミットメントと、応答と、挑戦値とから、前記入力文列から前記出力暗号文列への変換情報を保有していることを検証する、変換情報保有検証ステップと、
前記変換の満たす条件に関する変換条件コミットメントと、前記応答と、前記挑戦値とから、前記入力文列から前記出力暗号文列への変換の満たす条件を検証する、変換条件検証ステップと、
を含み、
前記変換情報保有検証ステップと前記変換条件検証ステップの検証がともに受理されたら、再暗号シャッフル検証結果として受理を、それ以外は不受理を出力する、ことを特徴とする再暗号シャッフル検証方法。
【請求項３】請求項１に記載の証明付再暗号シャッフル方法において、
前記変換情報保有コミットメント生成ステップは、前記出力暗号文列および前記変換情報保有コミットメントを、前記入力文列を基底として、再暗号化に用いた変数および並び替えに対応する値および乱数とを表現とした表現値として生成し、
前記変換条件コミットメント生成ステップは、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式の係数を、再暗号シャッフル情報から積和演算のみを用いて生成し、前記恒等式の係数、または前記係数の一部もしくは全てを、コミットしたものを前記変換条件コミットメントとし、

前記応答生成ステップは、前記応答を、再暗号シャッフル情報と挑戦値から積和演算のみで生成し、
前記表現は、表現値と基底を対応付けるものであって、無作為に与えられた表現値と基底から、これらを対応付ける表現を計算することは、計算量的に困難とされており、
前記挑戦値は、前記入力文列と前記出力暗号文列とコミットメント全てが決まった後に、無作為に決められる複数の成分、あるいは、前記入力文列と前記出力暗号文列と全てのコミットメントとを入力として挑戦値生成関数により出力される複数の成分であり、
前記挑戦値生成関数は、与えられた入力から複数の成分を出力するものであって、それらの出力からは入力を求めること、出力成分間の関係を意図して入力を決定することが、計算量的に困難である関数である、
ことを特徴とする証明付再暗号シャッフル方法。
【請求項４】請求項２に記載の再暗号シャッフル検証方法において、
前記変換情報保有検証ステップは、前記出力暗号文列および変換情報保有コミットメントを基底として挑戦値を表現とする表現値と、前記入力文列を基底として応答を表現とする表現値が等しいことを検証し、
前記変換条件検証ステップは、前記変換条件コミットメントにより、入力された応答と挑戦値に関して、前記入力文列から出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式が成立することを検証する、ことを特徴とする再暗号シャッフル検証方法。
【請求項５】請求項３に記載の証明付再暗号シャッフル方法において、
前記変換条件コミットメント生成ステップは、前記入力文列から前記出力暗号文列への変換の満たす条件を記述し、かつ、前記応答と準応答と挑戦値との多項式である恒等式の係数を、前記再暗号シャッフル情報から積和演算を用いて生成し、これらの恒等式の係数、または、これら係数の一部または全てをコミットしたものと、準元係数、または、これら係数の一部または全てをコミットしたものと、を変換条件コミットメントとして生成し、
前記準応答は、再暗号シャッフル検証における、変換情報保有検証処理には使用されないものであり、応答と挑戦値との多項式であり、前記多項式の係数が準元係数であり、
前記応答生成ステップは、前記挑戦値より再暗号シャッフル情報を用いて応答と準応答との二種の応答を生成し、
前記再暗号シャッフル証明文は、前記変換情報保有コミットメントと前記変換条件コミットメントと前記応答と前記準応答とを含む、
ことを特徴とする証明付再暗号シャッフル方法。
【請求項６】請求項４に記載の再暗号シャッフル検証方法において、
前記変換条件検証ステップは、前記変換条件コミットメントにより、入力された応答と準応答と挑戦値に関して、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と準応答と挑戦値の多項式である恒等式が成立することを検証し、準元係数をコミットしたものと応答と挑戦値とより、前記準応答の正当性を検証する、ことを特徴とする再暗号シャッフル検証方法。
【請求項７】請求項１に記載の証明付再暗号シャッフル方法において、
前記変換情報保有コミットメント生成ステップは、前記出力暗号文列および前記変換情報保有コミットメントを、前記入力文列を基底として、再暗号化に用いた変数および並び替えに対応する値および乱数とを表現とした表現値として生成し、
前記変換条件コミットメント生成ステップは、複数個の変換条件コミットメント生成ステップよりなり、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式の係数を、再暗号シャッフル情報から積和演算を用いて生成し、前記恒等式の係数、前記係数の一部もしくは全てを、コミットしたものを前記変換条件コミットメントとする変換条件コミットメント生成ステップと、
前記入力文列から前記出力暗号文列への変換の満たす条件を記述し、かつ、前記応答と準応答と挑戦値との多項式である恒等式の係数を、前記再暗号シャッフル情報から積和演算を用いて生成し、これらの恒等式の係数または、これら係数の一部または全てをコミットしたものと、準元係数または、これら係数の一部または全てをコミットしたものと、を変換条件コミットメントとして生成する変換条件コミットメント生成ステップと、の両ステップの一方または両方を複数含み、
前記応答生成ステップは、前記応答と、前記変換条件コミットメント生成処理に応じて複数個の準応答を生成し、
前記再暗号シャッフル証明文は、複数の前記変換条件コミットメントと、このコミットメントに対応する準応答と、前記応答と、前記変換情報保有コミットメントとを含む、ことを特徴とする証明付再暗号シャッフル方法。
【請求項８】請求項２に記載の再暗号シャッフル検証方法において、
前記変換情報保有検証ステップは、前記出力暗号文列および変換情報保有コミットメントを基底として挑戦値を表現とする表現値と、前記入力文列を基底として応答を表現とする表現値が等しいことを検証し、
前記変換条件検証ステップは、複数個の変換条件検証ステップよりなり、前記変換条件コミットメントにより、入力された応答と挑戦値に関して、前記入力文列から出力暗号文列への変換の満たす条件を記述する、応答と挑

戦値の多項式である恒等式が成立することを検証する検証ステップと、
前記変換条件コミットメントにより、入力された応答と準応答と挑戦値に関して、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と準応答と挑戦値の多項式である恒等式が成立することを検証し、準元係数をコミットしたものと応答と挑戦値とより、前記準応答の正当性を検証する検証ステップと、の両ステップの一方または両方を複数含む、ことを特徴とする再暗号シャッフル検証方法。
【請求項９】請求項１に記載の証明付再暗号シャッフル方法において、
前記変換情報保有コミットメント生成ステップは、複数個の変換情報保有コミットメント生成ステップよりなり、前記各変換情報保有コミットメント生成ステップは、前記出力暗号文列および前記変換情報保有コミットメントを、前記入力文列を基底として、再暗号化に用いた変数および並び替えに対応する値および乱数とを表現とした表現値として生成し、２番目以降の変換情報保有コミットメント生成処理は、１番目の変換情報保有コミットメント生成ステップと共通する出力の生成を省略し、
前記変換条件コミットメント生成ステップは、複数個の変換条件コミットメント生成ステップよりなり、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式の係数を、再暗号シャッフル情報から積和演算を用いて生成し、前記恒等式の係数、前記係数の一部もしくは全てを、コミットしたものを前記変換条件コミットメントとするステップと、
前記入力文列から前記出力暗号文列への変換の満たす条件を記述し、かつ、前記応答と準応答と挑戦値との多項式である恒等式の係数を、前記再暗号シャッフル情報から積和演算を用いて生成し、これらの恒等式の係数または、これら係数の一部または全てをコミットしたものと、準元係数または、これら係数の一部または全てをコミットしたものと、を変換条件コミットメントとして生成するステップと、の両ステップの一方または両方を複数含み、
前記応答生成ステップは、前記変換情報保有コミットメント生成ステップに応じて複数個の応答を生成し、前記変換条件コミットメント生成ステップに応じて複数個の準応答を生成し、
前記再暗号シャッフル証明文は、前記複数個の応答と複数個の知識のコミットメントと複数個の変換条件コミットメントとそれに対応する準応答とよりなるものである、ことを特徴とする証明付再暗号シャッフル方法。
【請求項１０】請求項２に記載の再暗号シャッフル検証方法において、
前記変換情報保有検証ステップは、複数個の変換情報保有検証ステップよりなり、前記各変換情報保有検証ステップは、前記出力暗号文列および変換情報保有コミットメントを基底として挑戦値を表現とする表現値と、前記入力文列を基底として応答を表現とする表現値が等しいことを検証し、
前記変換条件検証ステップは、複数個の変換条件検証ステップよりなり、前記変換条件コミットメントにより、入力された応答と挑戦値に関して、前記入力文列から出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式が成立することを検証するステップと、
前記変換条件コミットメントにより、入力された応答と準応答と挑戦値に関して、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と準応答と挑戦値の多項式である恒等式が成立することを検証し、準元係数をコミットしたものと応答と挑戦値とより、前記準応答の正当性を検証するステップと、の両ステップの一方または両方を複数含む、ことを特徴とする再暗号シャッフル検証方法。
【請求項１１】請求項３または請求項５に記載の証明付再暗号シャッフル方法において、
前記変換条件コミットメント生成ステップにおける恒等式が、応答の各成分は挑戦値の多項式よりなり、前記多項式の一部の、一部の項の各２乗の和と、挑戦値の成分の一部の各２乗の和とが、挑戦値によらずに等しくなる関係を内包している、ものであることを特徴とする証明付再暗号シャッフル方法。
【請求項１２】請求項３または請求項５に記載の証明付再暗号シャッフル方法において、
前記変換条件コミットメント生成ステップにおける恒等式が、応答の各成分は挑戦値の多項式であるが、これらの多項式の一部の、一部の項の各３乗の和と、挑戦値の成分の一部の各３乗の和とが、挑戦値によらずに等しくなる関係を内包しているものであることを特徴とする証明付再暗号シャッフル方法。
【請求項１３】請求項７または請求項９に記載の証明付再暗号シャッフル方法において、
前記変換条件コミットメント生成ステップにおける複数個の恒等式が、応答の各成分は挑戦値の多項式よりなり、前記多項式の一部の、一部の項の各２乗の和と、挑戦値の成分の一部の各２乗の和とが、挑戦値によらずに等しくなる関係を内包しているものと、応答の各成分は挑戦値の多項式であるが、これらの多項式の一部の、一部の項の各３乗の和と、挑戦値の成分の一部の各３乗の和とが、挑戦値によらずに等しくなる関係を内包しているものの二つを含む、ことを特徴とする証明付再暗号シャッフル方法。
【請求項１４】証明付再暗号シャッフル装置に入力する入力文列を、その一部を疑似乱数、または、公開鍵と入力暗号文列とに疑似乱数による変換を受けた数値として

生成する、ことを特徴とする入力文列生成方法。
【請求項１５】請求項１４に記載の入力文列生成方法において、入力暗号文列と公開鍵と疑似乱数を合わせて入力文列とする、ことを特徴とする入力文列生成方法。
【請求項１６】請求項１４に記載の入力文列生成方法において、公開鍵列により入力暗号文列を変換し、公開鍵列の正当性証明文を出力する入力文列生成方法であって、前記公開鍵列が、秘密鍵を分散所持する証明者が協力して生成した、同じ秘密鍵に対応する複数の公開鍵で、その各公開鍵の特定の成分からなる集合は、入力文列を含む入力により生成された疑似乱数である、ことを特徴とする入力文列生成方法。
【請求項１７】請求項１４に記載の入力文列生成方法において、秘密鍵を分散所持する証明者が協力して生成した同じ秘密鍵に対応する複数の公開鍵で、その各公開鍵の特定の成分からなる集合は、入力文列を含む入力により生成された疑似乱数である公開鍵列を生成し、これを構成する各公開鍵で各入力平文を暗号化し、かつそれぞれの公開鍵で暗号化したことを証明し、この入力暗号文列と公開鍵を合わせて入力文列を生成する、ことを特徴とする入力文列生成方法。
【請求項１８】与えられた入力から一意的に決定される疑似乱数数列を成分に持ち、かつ同じ秘密鍵に対応する、公開鍵を成分とする公開鍵列と、同じ秘密鍵に対応することの証明文とを、秘密鍵を分散所持する証明者が協力して生成する、ことを特徴とする証明付公開鍵列生成方法。
【請求項１９】複数の入力暗号文と一又は複数の公開鍵とからなる入力文列と、入力暗号文の並び替え方と、再暗号化に用いた変数と、乱数とを含む再暗号シャッフル情報と、を入力し、前記暗号文に対して順番の並び替えと前記公開鍵による再暗号化とを施した出力暗号文列と、再暗号シャッフル証明文とを出力する証明付再暗号シャッフル装置であって、
前記入力文列から出力暗号文列を生成するとともに、前記入力文列から出力暗号文への変換情報の保有に関するコミットメント（「変換情報保有コミットメント」という）を生成する変換情報保有コミットメント生成部と、
前記変換の満たす条件に関するコミットメント（「変換条件コミットメント」という）を生成する、変換条件コミットメント生成部と、
再暗号シャッフル情報と挑戦値とから応答を生成する応答生成部と、
を備え、
前記変換情報保有コミットメントと前記変換条件コミットメントと前記応答とを前記再暗号シャッフル証明文として出力する、ことを特徴とする証明付再暗号シャッフル装置。
【請求項２０】複数の入力暗号文と一又は複数の公開鍵とからなる入力文列と、再暗号シャッフル情報とを入力し、前記暗号文に対して順番の並び替えと前記公開鍵による再暗号化とを施した出力暗号文列と、再暗号シャッフル証明文とを出力する証明付再暗号シャッフル装置に入力される前記入力文列と、
前記証明付再暗号シャッフル装置から出力される前記出力暗号文列と、
前記証明付再暗号シャッフル装置から出力される、前記入力文列から前記出力暗号文への変換情報の保有に関する変換情報保有コミットメントと、前記変換を満する条件に関する変換条件コミットメントと、応答とを含む再暗号シャッフル証明文と、
を入力とし、受理または不受理である検証結果を出力する再暗号シャッフル検証装置であって、
前記入力文列と、前記出力暗号文列と、前記変換情報保有コミットメントと、応答と、挑戦値とに基づき、前記入力文列から前記出力暗号文列への変換情報を保有していることを検証する、変換情報保有検証部と、
前記変換条件コミットメントと、前記応答と、前記挑戦値とに基づき、前記入力文列から前記出力暗号文列への変換の満たす条件を検証する、変換条件検証部と、
を備え、
前記変換情報保有検証部と前記変換条件検証部における検証がともに受理された場合に、再暗号シャッフル検証結果として受理を出力し、それ以外は不受理を出力する、ことを特徴とする再暗号シャッフル検証装置。
【請求項２１】請求項１９に記載の証明付再暗号シャッフル装置において、
前記変換情報保有コミットメント生成部が、前記出力暗号文列および前記変換情報保有コミットメントを、前記入力文列を基底として、再暗号化に用いた変数、および並び替えに対応する値、および、乱数とを表現とした表現値として生成する手段を備え、
前記変換条件コミットメント生成部が、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式の係数を、前記再暗号シャッフル情報から、積和演算を用いて生成し、前記恒等式の係数、前記係数の一部もしくは全てをコミットしたものを、前記変換条件コミットメントとして出力する手段を備え、
前記応答生成部が、前記応答を、前記再暗号シャッフル情報と、前記入力文列と前記出力暗号文列とコミットメント全てが決まった後に、無作為に決められる複数の成分、あるいは、前記入力文列と前記出力暗号文列と全てのコミットメントとを入力として挑戦値生成関数により出力される複数の成分である挑戦値から積和演算を用いて生成する手段を備えたことを特徴とする証明付再暗号シャッフル装置。
【請求項２２】請求項２０に記載の再暗号シャッフル検証装置において、
前記変換情報保有検証部が、前記出力暗号文列および変

換情報保有コミットメントを基底として挑戦値を表現とする表現値と、前記入力文列を基底として応答を表現とする表現値が等しいことを検証する手段を備え、
前記変換条件検証部が、前記変換条件コミットメントにより、入力された応答と挑戦値に関して、前記入力文列から出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式が成立することを検証する手段を備えた、ことを特徴とする再暗号シャッフル検証装置。
【請求項２３】請求項２１に記載の証明付再暗号シャッフル装置において、
前記変換条件コミットメント生成部が、前記入力文列から前記出力暗号文列への変換の満たす条件を記述し、かつ、前記応答と準応答と挑戦値との多項式である恒等式の係数を、前記再暗号シャッフル情報から積和演算を用いて生成する手段と、
前記恒等式の係数、または、これら係数の一部または全てをコミットしたものと、準元係数、または、これら係数の一部または全てをコミットしたものと、を変換条件コミットメントとして生成する手段と、を備え、
前記準応答は、これは応答と挑戦値との多項式であり、前記多項式の係数が準元係数であり、
前記応答生成部は、前記挑戦値より再暗号シャッフル情報を用いて応答と準応答との二種の応答を生成する手段を備え、
前記再暗号シャッフル証明文が、前記変換情報保有コミットメントと前記変換条件コミットメントと前記応答と前記準応答よりなる、
ことを特徴とする証明付再暗号シャッフル装置。
【請求項２４】請求項２２に記載の再暗号シャッフル検証装置において、
前記変換条件検証部が、前記変換条件コミットメントにより、入力された応答と準応答と挑戦値に関して、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と準応答と挑戦値の多項式である恒等式が成立することを検証し、準元係数をコミットしたものと応答と挑戦値とより、前記準応答の正当性を検証する、ことを特徴とする再暗号シャッフル検証装置。
【請求項２５】請求項１９に記載の証明付再暗号シャッフル装置において、
前記変換情報保有コミットメント生成部が、前記出力暗号文列および前記変換情報保有コミットメントを、前記入力文列を基底として、再暗号化に用いた変数および並び替えに対応する値および乱数とを表現とした表現値として生成する手段を備え、
前記変換条件コミットメント生成部を複数備え、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式の係数を、再暗号シャッフル情報から積和演算を用いて生成し、前記恒等式の係数、前記係数の一部もしくは全てを、コミットしたものを前記変換条件コミットメントとする前記変換条件コミットメント生成部と、
前記入力文列から前記出力暗号文列への変換の満たす条件を記述し、かつ、前記応答と準応答と挑戦値との多項式である恒等式の係数を、前記再暗号シャッフル情報から積和演算を用いて生成し、これらの恒等式の係数または、これら係数の一部または全てをコミットしたものと、準元係数または、これら係数の一部または全てをコミットしたものと、を変換条件コミットメントとして生成する変換条件コミットメント生成部と、の両生成部の一方または両方を含み、
前記応答生成部が、前記応答と、前記複数の変換条件コミットメント生成部に応じて複数個の準応答を生成する手段を備え、
前記再暗号シャッフル証明文が、複数の変換条件コミットメントとそれに対応する準応答と、応答と変換情報保有コミットメントよりなる、ことを特徴とする証明付再暗号シャッフル装置。
【請求項２６】請求項２０に記載の再暗号シャッフル検証装置において、
前記変換情報保有検証部が、前記出力暗号文列および変換情報保有コミットメントを基底として挑戦値を表現とする表現値と、前記入力文列を基底として応答を表現とする表現値が等しいことを検証し、
前記変換条件検証部が、複数個の変換条件検証部よりなり、
前記変換条件コミットメントにより、入力された応答と挑戦値に関して、前記入力文列から出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式が成立することを検証する検証部と、
前記変換条件コミットメントにより、入力された応答と準応答と挑戦値に関して、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と準応答と挑戦値の多項式である恒等式が成立することを検証し、準元係数をコミットしたものと応答と挑戦値とより、前記準応答の正当性を検証する検証部と、の両検証部の一方または両方を複数含むことを特徴とする再暗号シャッフル検証装置。
【請求項２７】請求項１９に記載の証明付再暗号シャッフル装置において、
前記変換情報保有コミットメント生成部が、複数の変換情報保有コミットメント生成部よりなり、前記出力暗号文列および前記変換情報保有コミットメントを、前記入力文列を基底として、再暗号化に用いた変数および並び替えに対応する値および乱数とを表現とした表現値として生成し、２番目以降の変換情報保有コミットメント生成処理は１番目の変換情報保有コミットメント生成処理と共通する出力の生成を省略し、
前記変換条件コミットメント生成部が、複数の変換条件コミットメント生成部よりなり、前記入力文列から前記

出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式の係数を、再暗号シャッフル情報から積和演算を用いて生成し、前記恒等式の係数、前記係数の一部もしくは全てを、コミットしたものを前記変換条件コミットメントとする変換条件コミットメント生成部と、
前記入力文列から前記出力暗号文列への変換の満たす条件を記述し、かつ、前記応答と準応答と挑戦値との多項式である恒等式の係数を、前記再暗号シャッフル情報から積和演算を用いて生成し、これらの恒等式の係数または、これら係数の一部または全てをコミットしたものと、準元係数または、これら係数の一部または全てをコミットしたものと、を変換条件コミットメントとして生成する変換条件コミットメント生成部と、の両生成部の一方または両方を複数含み、
前記応答生成処理が、前記複数の変換情報保有コミットメント生成部の出力に応じて複数個の応答を生成し、前記複数の変換条件コミットメント生成部の出力に応じて複数個の準応答を生成する手段を備え、
前記再暗号シャッフル証明文は、前記複数個の応答と複数個の知識のコミットメントと複数個の変換条件コミットメントとそれに対応する準応答よりなる、ことを特徴とする証明付再暗号シャッフル装置。
【請求項２８】請求項２０に記載の再暗号シャッフル検証装置において、
前記変換情報保有検証部が、複数の変換情報保有検証部よりなり、前記出力暗号文列および変換情報保有コミットメントを基底として挑戦値を表現とする表現値と、前記入力文列を基底として応答を表現とする表現値が等しいことを検証し、
前記変換条件検証部が、複数の変換条件検証部よりなり、前記変換条件コミットメントにより、入力された応答と挑戦値に関して、前記入力文列から出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式が成立することを検証する変換条件検証部と、
前記変換条件コミットメントにより、入力された応答と準応答と挑戦値に関して、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と準応答と挑戦値の多項式である恒等式が成立することを検証し、準元係数をコミットしたものと応答と挑戦値とより、前記準応答の正当性を検証する変換条件検証部と、の両検証部の一方または両方を複数含むことを特徴とする再暗号シャッフル検証装置。
【請求項２９】請求項２１または請求項２３に記載の証明付再暗号シャッフル装置において、
前記変換条件コミットメント生成部における恒等式が、応答の各成分は、挑戦値の多項式よりなり、前記多項式の一部の、一部の項の各２乗の和と、挑戦値の成分の一部の各２乗の和とが、挑戦値によらずに等しくなる関係を内包している、ものであることを特徴とする証明付再暗号シャッフル装置。
【請求項３０】請求項２１または請求項２３に記載の証明付再暗号シャッフル装置において、
前記変換条件コミットメント生成部における恒等式が、応答の各成分は挑戦値の多項式であるが、これらの多項式の一部の、一部の項の各３乗の和と、挑戦値の成分の一部の各３乗の和とが、挑戦値によらずに等しくなる関係を内包しているものである、ことを特徴とする証明付再暗号シャッフル装置。
【請求項３１】請求項２５または請求項２７に記載の証明付再暗号シャッフル装置において、
前記変換条件コミットメント生成部における複数個の恒等式が、応答の各成分は挑戦値の多項式よりなり、前記多項式の一部の、一部の項の各２乗の和と、挑戦値の成分の一部の各２乗の和とが、挑戦値によらずに等しくなる関係を内包しているものと、応答の各成分は挑戦値の多項式であるが、これらの多項式の一部の、一部の項の各３乗の和と、挑戦値の成分の一部の各３乗の和とが、挑戦値によらずに等しくなる関係を内包しているもののを含む、ことを特徴とする再暗号シャッフル装置。
【請求項３２】請求項１９記載の前記証明付再暗号シャッフル装置に入力する前記入力文列を、その一部を、疑似乱数または、公開鍵と入力暗号文列とに疑似乱数による変換を受けた数値として生成する入力文列生成装置。
【請求項３３】請求項３２に記載の入力文列生成装置において、入力暗号文列と公開鍵と疑似乱数を合わせて入力文列とする入力文列生成装置。
【請求項３４】請求項３２に記載の入力文列生成装置において、公開鍵列により入力暗号文列を変換し、公開鍵列の正当性証明文を出力する入力文列生成装置であって、前記公開鍵列が、秘密鍵を分散所持する証明者が協力して生成した、同じ秘密鍵に対応する複数の公開鍵で、その各公開鍵の特定の成分からなる集合は、入力文列を含む入力により生成された疑似乱数である、ことを特徴とする入力文列生成装置。
【請求項３５】請求項３２に記載の入力文列生成装置において、秘密鍵を分散所持する証明者が協力して生成した同じ秘密鍵に対応する複数の公開鍵で、その各公開鍵の特定の成分からなる集合は、入力文列を含む入力により生成された疑似乱数である公開鍵列を生成し、これを構成する各公開鍵で各入力平文を暗号化し、かつそれぞれの公開鍵で暗号化したことを証明し、この入力暗号文列と公開鍵を合わせて入力文列を生成する入力文列生成装置。
【請求項３６】与えらた入力から一意的に決定される疑似乱数数列を成分に持ち、かつ同じ秘密鍵に対応する、公開鍵を成分とする公開鍵列と、同じ秘密鍵に対応することの証明文とを、秘密鍵を分散所持する証明者が協力して生成する手段を備えたことを特徴とする証明付公開

鍵列生成装置。
【請求項３７】複数の入力暗号文と一又は複数の公開鍵とからなる入力文列と、入力暗号文の並び替え方と、再暗号化に用いた変数と、乱数とを含む再暗号シャッフル情報と、を入力し、前記暗号文に対して、順番の並び替えと、前記公開鍵による再暗号化を施した出力暗号文列と、再暗号シャッフル証明文とを出力する再暗号シャッフル装置であって、
（ａ）前記入力文列から出力暗号文列を生成するとともに、前記入力文列から出力暗号文への変換情報の保有に関するコミットメント（「変換情報保有コミットメント」という）を生成する変換情報保有コミットメント生成処理と、
（ｂ）前記変換の満たす条件に関するコミットメント（「変換条件コミットメント」という）を生成する、変換条件コミットメント生成処理と、
（ｃ）再暗号シャッフル情報と挑戦値とから応答を生成する応答生成処理と、
（ｄ）前記変換情報保有コミットメントと前記変換条件コミットメントと前記応答とを前記再暗号シャッフル証明文として出力する処理、
の前記（ａ）乃至（ｄ）の処理を証明付再暗号シャッフル装置を構成するコンピュータで実行させるためのプログラムを記録した記録媒体。
【請求項３８】入力文列と、証明付再暗号シャッフル装置から出力される出力暗号文列と、再暗号シャッフル装置から出力される、前記入力文列から前記出力暗号文への変換情報の保有に関する変換情報保有コミットメントと、前記変換を満する条件に関する変換条件コミットメントと、応答とを含む再暗号シャッフル証明文と、を入力とし、受理または不受理である検証結果を出力する再暗号シャッフル検証装置であって、
（ａ）前記入力文列と、前記出力暗号文列と、前記変換情報保有コミットメントと、応答と、挑戦値とより、前記入力文列から前記出力暗号文列への変換情報を保有していることを検証する、変換情報保有検証処理と、
（ｂ）前記変換条件コミットメントと、前記応答と、前記挑戦値とから、前記入力文列から前記出力暗号文列への変換の満たす条件を検証する、変換条件検証処理と、
（ｃ）前記変換情報保有検証処理と前記変換条件検証処理がともに受理された場合に、再暗号シャッフル検証結果として受理を出力し、それ以外は不受理を出力する処理、
の前記（ａ）乃至（ｃ）の処理を再暗号シャッフル検証装置を構成するコンピュータで実行させるためのプログラムを記録した記録媒体。
【請求項３９】請求項３７に記載の記憶媒体において、
前記変換情報保有コミットメント生成処理が、
前記出力暗号文列および前記変換情報保有コミットメントを、前記入力文列を基底として、再暗号化に用いた変数、および並び替えに対応する値、および、乱数とを表現とした表現値として生成し、
前記変換条件コミットメント生成処理が、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式の係数を、前記再暗号シャッフル情報から、積和演算を用いて生成し、
前記恒等式の係数、前記係数の一部もしくは全てをコミットしたものを、前記変換条件コミットメントとして出力し、
前記応答生成処理は、前記応答を、前記再暗号シャッフル情報と、前記入力文列と前記出力暗号文列とコミットメント全てが決まった後に、無作為に決められる複数の成分、あるいは、前記入力文列と前記出力暗号文列と全てのコミットメントとを入力として挑戦値生成関数により出力される複数の成分である挑戦値から積和演算を用いて生成する、
前記各処理を前記証明付再暗号シャッフル装置を構成するコンピュータで実行させるためのプログラムを記録した記録媒体。
【請求項４０】請求項３８に記載の記憶媒体において、
前記変換情報保有検証処理は、前記出力暗号文列および変換情報保有コミットメントを基底として挑戦値を表現とする表現値と、前記入力文列を基底として応答を表現とする表現値が等しいことを検証し、
前記変換条件検証処理は、前記変換条件コミットメントにより、入力された応答と挑戦値に関して、前記入力文列から出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式が成立することを検証する、
前記各処理を前記再暗号シャッフル検証装置を構成するコンピュータで実行させるためのプログラムを記録した記録媒体。
【請求項４１】請求項３７に記載の記憶媒体において、
前記変換条件コミットメント生成処理が、前記入力文列から前記出力暗号文列への変換の満たす条件を記述し、かつ、前記応答と準応答と挑戦値との多項式である恒等式の係数を、前記再暗号シャッフル情報から積和演算を用いて生成し、
前記恒等式の係数、または、これら係数の一部または全てをコミットしたものと、準元係数、または、これら係数の一部または全てをコミットしたものと、を変換条件コミットメントとして生成し、
前記準応答は、これは応答と挑戦値との多項式であり、前記多項式の係数が準元係数であり、
前記応答生成処理は、前記挑戦値より再暗号シャッフル情報を用いて応答と準応答との二種の応答を生成し、
前記再暗号シャッフル証明文として、前記変換情報保有コミットメントと前記変換条件コミットメントと前記応答と前記準応答を出力する、
前記各処理を前記証明付再暗号シャッフル装置を構成す

るコンピュータで実行させるためのプログラムを記録した記録媒体。
【請求項４２】請求項４０に記載の記録媒体において、前記変換条件検証処理が、前記変換条件コミットメントにより、入力された応答と準応答と挑戦値に関して、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と準応答と挑戦値の多項式である恒等式が成立することを検証し、準元係数をコミットしたものと応答と挑戦値とより、前記準応答の正当性を検証する、
前記変換条件検証処理を、前記再暗号シャッフル検証装置を構成するコンピュータで実行させるためのプログラムを記録した記録媒体。
【請求項４３】請求項３７に記載の記録媒体において、前記変換情報保有コミットメント生成処理が、前記出力暗号文列および前記変換情報保有コミットメントを、前記入力文列を基底として、再暗号化に用いた変数および並び替えに対応する値および乱数とを表現とした表現値として生成し、
複数の前記変換条件コミットメント生成処理を備え、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式の係数を、再暗号シャッフル情報から積和演算を用いて生成し、前記恒等式の係数、前記係数の一部もしくは全てを、コミットしたものを前記変換条件コミットメントとする前記変換条件コミットメント生成処理と、
前記入力文列から前記出力暗号文列への変換の満たす条件を記述し、かつ、前記応答と準応答と挑戦値との多項式である恒等式の係数を、前記再暗号シャッフル情報から積和演算を用いて生成し、これらの恒等式の係数または、これら係数の一部または全てをコミットしたものと、準元係数または、これら係数の一部または全てをコミットしたものと、を変換条件コミットメントとして生成する変換条件コミットメント生成処理と、をからなり、
前記応答生成処理は、前記応答と、前記複数の変換条件コミットメント生成部に応じて複数個の準応答を生成し、
前記再暗号シャッフル証明文として複数の変換条件コミットメントとそれに対応する準応答と、応答と変換情報保有コミットメントを出力する、
前記各処理を、前記証明付再暗号シャッフル装置を構成するコンピュータで実行させるためのプログラムを記録した記録媒体。
【請求項４４】請求項３８に記載の記録媒体において、前記変換情報保有検証処理が、前記出力暗号文列および変換情報保有コミットメントを基底として挑戦値を表現とする表現値と、前記入力文列を基底として応答を表現とする表現値が等しいことを検証し、
前記変換条件検証処理が、複数の変換条件検証処理よりなり、
前記変換条件コミットメントにより、入力された応答と挑戦値に関して、前記入力文列から出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式が成立することを検証する検証処理と、
前記変換条件コミットメントにより、入力された応答と準応答と挑戦値に関して、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と準応答と挑戦値の多項式である恒等式が成立することを検証し、準元係数をコミットしたものと応答と挑戦値とより、前記準応答の正当性を検証する検証処理と、からなり、
前記各処理を、前記再暗号シャッフル検証装置を構成するコンピュータで実行させるためのプログラムを記録した記録媒体。
【請求項４５】請求項３７に記載の記録媒体において、前記変換情報保有コミットメント生成処理が、複数の変換情報保有コミットメント生成処理よりなり、前記出力暗号文列および前記変換情報保有コミットメントを、前記入力文列を基底として、再暗号化に用いた変数および並び替えに対応する値および乱数とを表現とした表現値として生成し、２番目以降の変換情報保有コミットメント生成処理は１番目の変換情報保有コミットメント生成処理と共通する出力の生成を省略し、
前記変換条件コミットメント生成処理が、複数の変換条件コミットメント生成処理よりなり、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式の係数を、再暗号シャッフル情報から積和演算を用いて生成し、前記恒等式の係数、前記係数の一部もしくは全てを、コミットしたものを前記変換条件コミットメントとする変換条件コミットメント生成処理と、前記入力文列から前記出力暗号文列への変換の満たす条件を記述し、かつ、前記応答と準応答と挑戦値との多項式である恒等式の係数を、前記再暗号シャッフル情報から積和演算を用いて生成し、これらの恒等式の係数または、これら係数の一部または全てをコミットしたものと、準元係数または、これら係数の一部または全てをコミットしたものと、を変換条件コミットメントとして生成する変換条件コミットメント生成処理と、を含み、
前記応答生成処理は、前記複数の変換情報保有コミットメント生成部の出力に応じて複数個の応答を生成し、前記複数の変換条件コミットメント生成部の出力に応じて複数個の準応答を生成し、
前記再暗号シャッフル証明文として前記複数個の応答と複数個の知識のコミットメントと複数個の変換条件コミットメントとそれに対応する準応答とを出力する、前記各処理を、前記証明付再暗号シャッフル装置を構成するコンピュータで実
行させるためのプログラムを記録した記録媒体。

【請求項４６】請求項３８に記載の記録媒体において、前記変換情報保有検証処理が、複数の変換情報保有検証処理よりなり、前記出力暗号文列および変換情報保有コミットメントを基底として挑戦値を表現とする表現値と、前記入力文列を基底として応答を表現とする表現値が等しいことを検証し、
前記変換条件検証処理が、複数の変換条件検証処理よりなり、前記変換条件コミットメントにより、入力された応答と挑戦値に関して、前記入力文列から出力暗号文列への変換の満たす条件を記述する、応答と挑戦値の多項式である恒等式が成立することを検証する変換条件検証処理と、
前記変換条件コミットメントにより、入力された応答と準応答と挑戦値に関して、前記入力文列から前記出力暗号文列への変換の満たす条件を記述する、応答と準応答と挑戦値の多項式である恒等式が成立することを検証し、準元係数をコミットしたものと応答と挑戦値とより、前記準応答の正当性を検証する変換条件検証処理と、を含み、
前記各処理を、前記再暗号シャッフル検証装置を構成するコンピュータで実行させるためのプログラムを記録した記録媒体。
【請求項４７】請求項３９または請求項４１に記載の記録媒体において、
前記変換条件コミットメント生成処理における恒等式が、応答の各成分は、挑戦値の多項式よりなり、前記多項式の一部の、一部の項の各２乗の和と、挑戦値の成分の一部の各２乗の和とが、挑戦値によらずに等しくなる関係を内包しており、
前記変換条件コミットメント生成処理を、前記証明付再暗号シャッフル装置を構成するコンピュータで実行させるためのプログラムを記録した記録媒体。
【請求項４８】請求項３９または請求項４１に記載の記録媒体において、
前記変換条件コミットメント生成処理における恒等式が、応答の各成分は挑戦値の多項式であるが、これらの多項式の一部の、一部の項の各３乗の和と、挑戦値の成分の一部の各３乗の和とが、挑戦値によらずに等しくなる関係を内包しており、
前記変換条件コミットメント生成処理を、前記証明付再暗号シャッフル装置を構成するコンピュータで実行させるためのプログラムを記録した記録媒体。
【請求項４９】請求項４３または請求項４５に記載の記録媒体において、
前記変換条件コミットメント生成処理における複数個の恒等式が、応答の各成分は挑戦値の多項式よりなり、前記多項式の一部の、一部の項の各２乗の和と、挑戦値の成分の一部の各２乗の和とが、挑戦値によらずに等しくなる関係を内包しているものと、応答の各成分は挑戦値の多項式であるが、これらの多項式の一部の、一部の項の各３乗の和と、挑戦値の成分の一部の各３乗の和とが、挑戦値によらずに等しくなる関係を内包しているものの二つを含み、
前記変換条件コミットメント生成処理を、前記証明付再暗号シャッフル装置を構成するコンピュータで実行させるためのプログラムを記録した記録媒体。
【請求項５０】請求項３７に記載の記録媒体において、前記証明付再暗号シャッフル装置に入力する前記入力文列を、その一部を、疑似乱数または、公開鍵と入力暗号文列とに疑似乱数による変換を受けた数値として生成する入力文列生成処理を、コンピュータで実行させるためのプログラムを記録した記録媒体。
【請求項５１】請求項５０に記載の記録媒体において、入力暗号文列と公開鍵と疑似乱数を合わせて入力文列とする入力文列生成処理をコンピュータで実行させるためのプログラムを記録した記録媒体。
【請求項５２】請求項５０に記載の記録媒体において、公開鍵列により入力暗号文列を変換し、公開鍵列の正当性証明文を出力する入力文列生成処理であって、前記公開鍵列が、秘密鍵を分散所持する証明者が協力して生成した、同じ秘密鍵に対応する複数の公開鍵であり、前記各公開鍵の特定の成分からなる集合は、入力文列を含む入力により生成された疑似乱数とする、入力文列生成処理をコンピュータで実行させるためのプログラムを記録した記録媒体。
【請求項５３】請求項５０に記載の記録媒体において、秘密鍵を分散所持する証明者が協力して生成した同じ秘密鍵に対応する複数の公開鍵で、その各公開鍵の特定の成分からなる集合は、入力文列を含む入力により生成された疑似乱数である公開鍵列を生成し、これを構成する各公開鍵で各入力平文を暗号化し、かつそれぞれの公開鍵で暗号化したことを証明し、この入力暗号文列と公開鍵を合わせて入力文列を生成する入力文列生成処理をコンピュータで実行させるためのプログラムを記録した記録媒体。
【請求項５４】与えらた入力から一意的に決定される疑似乱数数列を成分に持ち、かつ同じ秘密鍵に対応する、公開鍵を成分とする公開鍵列と、同じ秘密鍵に対応することの証明文とを、秘密鍵を分散所持する証明者が協力して生成する証明付公開鍵列生成処理を、コンピュータで実行させるためのプログラムを記録した記録媒体。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】本発明は、匿名通信路の構成などに使われる、入出力暗号文の一対一の対応関係を秘匿しつつ一対一の対応の存在を保証する再暗号シャッフル技術および再暗号シャッフル検証技術に関する。
【０００２】
【従来の技術】［従来の技術(1)］従来の証明付再暗号シャッフルの技術について、例えば特開平08-263575号

公報（「文献１」という）の記載が参照される。図１に、該文献１に記載される構成を示す。なお、本願添付図面中で、合流する矢印は、矢印の元の情報が、全て集まって矢印の先へ送られることを意味し、分岐する矢印は矢印の元の情報全てまたは一部が、それぞれの矢印の先へ送られることを意味する。また破線は使用する入力文列生成法に依存することを示す。

【０００３】図１において、１６０個ある疑似出力暗号文列は零知識証明のコミットメントである。挑戦値は、入出力暗号文とコミットメントより生成して、応答は挑戦値のビット値に応じて実線または点線の矢印で示される入力暗号文列または出力暗号文列から疑似出力暗号文列への写像の明示である。

【０００４】図１に示すように、複数のElgamal入力暗号100の順序を並び替えて再び暗号化して出力する手法が紹介されている。このような処理を、「暗号シャッフル」という。この処理が正当であることを保証する為に同文献では以下の手法が紹介されている。並べ替えと再暗号の秘密乱数を毎回異なるものにして、再暗号シャッフルと同様の操作を安全変数(約160)の回数繰り返して疑似出力暗号文列を出力し、これを正当性証明のコミットメントとする。そしてこれら入出力暗号文とコミットメントのハッシュ値を挑戦値105として生成する。

【０００５】この挑戦値のビット列を上から順に読み、ビットが“1”の時は入力暗号文列から、“0”の時は出力暗号文列からの並び替え(並び替えを表す写像)と再暗号化(再び暗号化した時に使ったの乱数)の明示を応答106とする。

【０００６】以上のコミットメント、挑戦値、応答を再暗号シャッフルの証明文として出力する。以上においてハッシュ値のビット値に応じて対応関係を明示する方法をCut&Choose（カット・アンド・チューズ）法と言う。

【０００７】［従来の技術(２)］他の従来技術としては、阿部が、1999年電子情報通信学会情報セキュリティ技術報告書で発表した、「AMix-network on Permutation Networks」（「文献２」という）が参照される。この文献２では、例えば図２に示すように、一対の入力暗号文の置換200を繰り返して、全体として複数の入力暗号文の並び替えを実現する。各入力暗号文の置換の証明を、Cut&Choose法でない方法で構成することにより、ある数より小さい入力暗号文数の証明付再暗号シャッフルとしては効率の向上を達成している。すなわち、個々の入力暗号文を置換することによって入力暗号文列全体の並び替えを実現している。個々の置換の証明は効率の良いものであるが、置換を多くそろえる必要がある。

【０００８】

【発明が解決しようとする課題】しかしながら、上記した従来の技術は下記記載の問題点を有している。

【０００９】従来の技術(1)においては、コミットメントの生成のために安全変数(約160)の回数暗号シャッフルしなければならない。一回の再暗号シャッフルは、入力暗号文の数の2倍の冪乗剰余演算を行わねばならず、計算量が多い。

【００１０】また検証は、コミットメントを生成するのと同数の冪乗剰余演算を行わねばならず、計算量が多い。

【００１１】次に従来の技術(2)においては、一対の入力暗号文の置換とその証明のコミットメントは、合わせて、８回の冪乗剰余演算が必要である。

【００１２】この一置換当りの計算量は、従来の技術(1)の２入力暗号文あたりの計算量(=320)と比較すると小さいものの、入力暗号文全体のどのような並び替えでも実現できる回数にわたる一対の入力暗号文の置換が必要とされており、この数は、入力暗号文の数をnとすると、n logn-n+1である。

【００１３】このため、入力暗号文の数が増大すると、計算量が多くなる。

【００１４】したがって、本発明は、上記問題点に鑑みてなされたものであって、その目的は、入力暗号文数に依存せずに証明の計算量の短縮を図る方法及びシステム並びに記録媒体を提供することにある。

【００１５】本発明の他の目的は、検証処理を、証明同様に計算量の短縮を図る方法及びシステム並びに記録媒体を提供することにある。これ以外の本発明の目的、特徴、利点等は以下の実施の形態の記載から、当業者には直ちに明らかとされるであろう。

【００１６】

【課題を解決するための手段】前記目的を達成する本発明による証明付再暗号シャッフル方法は、複数の暗号文と一つまたは複数の公開鍵とからなる入力文列と、再暗号シャッフル情報とを入力し、前記暗号文に対して、順番の並び替えと、前記公開鍵による再暗号化を施した出力暗号文列と、上記処理に関する証明文である再暗号シャッフル証明文とを出力する証明付再暗号シャッフル方法において、前記入力文列から出力暗号文列を生成するとともに、前記入力文列から出力暗号文への変換情報の保有に関するコミットメント（「変換情報保有コミットメント」という）を生成する変換情報保有コミットメント生成ステップと、前記変換の満たす条件に関するコミットメント（「変換条件コミットメント」という）を生成する、変換条件コミットメント生成ステップと、再暗号シャッフル情報と挑戦値とから応答を生成する、応答生成ステップと、を含み、前記変換情報保有コミットメントと前記変換条件コミットメントと前記応答とを前記再暗号シャッフル証明文として出力し、前記再暗号シャッフル情報は、入力暗号文の並び替え方と、再暗号化に用いた変数と、乱数とからなる、ことを特徴とする。

【００１７】また本発明に係る再暗号シャッフル検証方法は、入力文列と、出力暗号文列と、再暗号シャッフル証明文が入力され、受理または不受理である検証結果を

出力する再暗号シャッフル検証方法において、前記入力文列と、前記出力暗号文列と、変換情報保有コミットメントと、応答と、挑戦値とより、前記入力文列から前記出力暗号文列への変換情報を保有していることを検証する、変換情報保有検証ステップと、変換条件コミットメントと、前記応答と、前記挑戦値とから、前記入力文列から前記出力暗号文列への変換の満たす条件を検証する、変換条件検証ステップと、を含み、前記変換情報保有検証処理と前記変換条件検証処理の検証がともに受理されたら、再暗号シャッフル検証結果として受理を、それ以外は不受理を出力する、ことを特徴とする。

【００１８】本発明の証明付再暗号シャッフル装置は、複数の入力暗号文と一又は複数の公開鍵とからなる入力文列と、入力暗号文の並び替え方と、再暗号化に用いた変数と、乱数とを含む再暗号シャッフル情報と、を入力し、前記暗号文の順番を並び替え、前記公開鍵による再暗号化を施した出力暗号文列と、再暗号シャッフル証明文とを出力する証明付再暗号シャッフル装置であって、前記入力文列から出力暗号文列を生成するとともに、前記入力文列から出力暗号文への変換情報の保有に関するコミットメント（「変換情報保有コミットメント」という）を生成する変換情報保有コミットメント生成部と、前記変換の満たす条件に関するコミットメント（「変換条件コミットメント」という）を生成する、変換条件コミットメント生成部と、再暗号シャッフル情報と挑戦値とから応答を生成する応答生成部と、を備え、前記変換情報保有コミットメントと前記変換条件コミットメントと前記応答とを前記再暗号シャッフル証明文として出力する。

【００１９】本発明の再暗号シャッフル検証装置は、複数の入力暗号文と一又は複数の公開鍵とからなる入力文列と、入力暗号文の並び替え方と、再暗号化に用いた変数と、乱数とを含む再暗号シャッフル情報と、を入力し、前記暗号文の順番を並び替え、前記公開鍵による再暗号化を施した出力暗号文列と、再暗号シャッフル証明文とを出力する証明付再暗号シャッフル装置に入力される前記入力文列と、前記再暗号シャッフル装置から出力される前記出力暗号文列と、前記再暗号シャッフル装置から出力される前記再暗号シャッフル証明文とを入力とし、受理または不受理である検証結果を出力する再暗号シャッフル検証装置であって、前記入力文列と、前記出力暗号文列と、変換情報保有コミットメントと、応答と、挑戦値とより、前記入力文列から前記出力暗号文列への変換情報を保有していることを検証する、変換情報保有検証部と、前記変換条件コミットメントと、前記応答と、前記挑戦値とから、前記入力文列から前記出力暗号文列への変換の満たす条件を検証する、変換条件検証部と、を備え、前記変換情報保有検証部と前記変換条件検証部における検証がともに受理された場合に、再暗号シャッフル検証結果として受理を出力し、それ以外は不受理を出力する。

【００２０】また本発明に係る入力文列生成方法は、証明付再暗号シャッフル装置に入力する入力文列を、その一部を疑似乱数または、公開鍵と入力暗号文列とに疑似乱数による変換を受けた数値として生成する。本発明においては、入力暗号文列と公開鍵と疑似乱数を合わせて入力文列としてもよい。

【００２１】本発明に係る記録媒体は、複数の入力暗号文と一又は複数の公開鍵とからなる入力文列と、入力暗号文の並び替え方と、再暗号化に用いた変数と、乱数とを含む再暗号シャッフル情報と、を入力し、前記暗号文に対して、順番の並び替えと、前記公開鍵による再暗号化を施した出力暗号文列と、再暗号シャッフル証明文とを出力する証明付再暗号シャッフル装置であって、

（ａ）前記入力文列から出力暗号文列を生成するとともに、前記入力文列から出力暗号文への変換情報の保有に関するコミットメント（「変換情報保有コミットメント」という）を生成する変換情報保有コミットメント生成処理と、（ｂ）前記変換の満たす条件に関するコミットメント（「変換条件コミットメント」という）を生成する、変換条件コミットメント生成処理と、（ｃ）再暗号シャッフル情報と挑戦値とから応答を生成する応答生成処理と、（ｄ）前記変換情報保有コミットメントと前記変換条件コミットメントと前記応答とを前記再暗号シャッフル証明文として出力する処理、の前記（ａ）乃至（ｄ）の処理を再暗号シャッフル装置を構成するコンピュータで実行させるためのプログラムを記録している。

【００２２】本発明に係る記録媒体は、入力文列と、前記証明付再暗号シャッフル装置から出力される前記出力暗号文列と、前記証明付再暗号シャッフル装置から出力される、前記入力文列から前記出力暗号文への変換情報の保有に関する変換情報保有コミットメントと、前記変換の満たす条件に関する変換条件コミットメントと、前記応答とからなる再暗号シャッフル証明文とを入力とし、受理または不受理である検証結果を出力する再暗号シャッフル検証装置であって、（ａ）前記入力文列と、前記出力暗号文列と、前記変換情報保有コミットメントと、応答と、挑戦値とより、前記入力文列から前記出力暗号文列への変換情報を保有していることを検証する、変換情報保有検証処理と、（ｂ）前記変換条件コミットメントと、前記応答と、前記挑戦値とから、前記入力文列から前記出力暗号文列への変換の満たす条件を検証する、変換条件検証処理と、（ｃ）前記変換情報保有検証処理と前記変換条件検証処理がともに受理された場合に、再暗号シャッフル検証結果として受理を出力し、それ以外は不受理を出力する処理、の前記（ａ）乃至（ｃ）の処理を再暗号シャッフル検証装置を構成するコンピュータで実行させるためのプログラムを記録している。

【００２３】［発明の概要］本発明は、再暗号シャッフ

ルを、より一般的な変換の一種として表現し、この変換の情報を保有していることの証明と、この変換の満たす条件の証明との二つを合わせて、再暗号シャッフルの証明を構成している。

【００２４】これら二種の証明それぞれは、従来の再暗号シャッフルの証明よりも単純であり、入力暗号文数に依存せずに、証明の計算量が短縮されたものであり、また、これらを合わせた再暗号シャッフルの証明でも、この優位性が保たれる。

【００２５】変換の情報を保有していることの証明は、出力暗号文列と変換情報保有コミットメントを生成した後、挑戦値から上記変換と、変換情報保有コミットメント生成に使用する乱数とに依存して、応答を生成することにより行う。

【００２６】ここで、応答と挑戦値の関係に変換が反映されるため、変換の満たす条件から挑戦値に依存せずに、応答と挑戦値の満たす関係式が存在する。この関係式をコミットして変換の満たす条件を証明する。

【００２７】証明すべき変換の満たす条件として、再暗号シャッフルを表わす変換の満たす条件を選べば、両証明をもってして、再暗号シャッフルの証明を構成することができる。

【００２８】

【発明の実施の形態】本発明の上記および他の目的、特徴および利点を明確にすべく、以下添付した図面を参照しながら本発明の実施の形態につき詳細に説明する。

【００２９】最初に前提となる事柄について述べる。本発明において用いられる暗号方法は、確率暗号である公開鍵暗号系に属する暗号方法である。例えば、Elgamal暗号、楕円暗号、代数曲線暗号などもこれに含まれる。

【００３０】本発明に係る証明付再暗号シャッフル方法は、入力暗号文の作成者全員が証明付再暗号シャッフルを行う証明者に対して、入力暗号文の制作に用いた秘密変数を漏らさなければ、証明者は再暗号シャッフル証明文を偽造できないものである。ただし、本発明に係る入力文列生成方法を併せて用いることで、入力暗号文の作成者が証明者と共謀した場合においても、証明文の偽造を防止する、ことができる。

【００３１】本発明に係る証明付再暗号シャッフル方法は、変換情報保有コミットメントを生成する変換情報保有コミットメント生成処理と、変換条件コミットメントを生成する変換条件コミットメント生成処理と、応答や準応答を生成する応答生成処理とよりなり、証明文はこの上記３種類の処理より生成されるコミットメントと応答(応答および準応答)とよりなる。

【００３２】本発明に係る再暗号シャッフル検証方法は、入力文列と出力暗号文列と変換情報保有コミットメントと応答とから変換の情報を保有していることを検証する変換情報保有検証処理と、変換条件コミットメントと応答と準応答とから変換の満たす条件を検証する変換条件検証処理とよりなる。

【００３３】［変換情報保有コミットメント生成処理］証明付再暗号シャッフル方法を構成する変換情報保有コミットメント生成処理について説明する。

【００３４】変換情報保有コミットメント生成処理は、入力文列から再暗号シャッフルに対応する変換を行って出力暗号文列を生成し、入力文列から乱数による一般の変換を行って変換情報保有コミットメントを生成する。

【００３５】また入力文列に入力暗号文列と公開鍵以外が含まれている場合には、それから再暗号シャッフルに対応する変換を行ったものも変換情報保有コミットメントとする。

【００３６】応答を複数個生成する場合には、異る乱数による一般の変換を、複数個行い、変換情報保有コミットメントを、複数組生成する。

【００３７】例えば、この変換を、出力暗号文列および変換情報保有コミットメントを、入力文列を基底として、再暗号化に用いた変数および乱数および並び替えに対応する値とを表現とした表現値として生成できる。

【００３８】また、この表現とは、基底とある表現値を対応付けるものであり、かつ、基底と表現値から表現を計算することが計算量的に困難となる方法である必要があり、この表現方法に、冪乗剰余を用いることができる。

【００３９】例えば、入力暗号文列を、$g[i,\Gamma];i=1,\ldots,n;\Gamma=0,\ldots,l$、公開鍵を、$g[i,\Gamma];i=n+1,\ldots,n+m;\Gamma=0,\ldots,l$、それ以外の入力文列の成分を、$g[i,\Gamma];i=1,\ldots,n+m;\Gamma=l+1,\ldots,l'$、一般の変換に対応する乱数(以下「情報隠蔽因子」と呼ぶ)を、$A[\mu,j];\mu=1,\ldots,n+m,j=n+1,\ldots,n+m'$、再暗号化の変数を、$A[i,j];i=n+1,\ldots,n+m,j=1,\ldots,n$、並び替えに対応する変換を表す変数を、$A[i,j];i,j=1,\ldots,n$として、出力暗号文列　$g''[i,\Gamma];i=1,\ldots,n;\Gamma=1,\ldots,l$を、

$$g''_{A[j,i]}[i,\Gamma]=\Pi_{j=1}^{n}g[j,\Gamma]^{A[j,i]}\ \Pi_{j=n+1}^{n+m}g[j,\Gamma]\ /F^{*}_{p},\quad i=1,\ldots,n\ \Gamma=1,\ldots,l$$

と生成し、変換情報保有コミットメントを、

$$g''_{A[j,i]}[i,\Gamma]=\Pi_{j=1}^{n}g[j,\Gamma]^{A[j,i]}\ \Pi_{j=n+1}^{n+m}g[j,\Gamma]\ /F^{*}_{p},\quad i=n+1,\ldots,n+m'\quad \Gamma=1,\ldots,l$$

と生成し、入力文列に$g[i,\Gamma];i=1,\ldots,n+m;\Gamma=l+1,\ldots,l'$が含まれた場合の変換情報保有コミットメントを、

$$g''_{A[j,i]}[i,\Gamma]=\Pi_{j=1}^{n}g[j,\Gamma]^{A[j,i]}\ \Pi_{j=n+1}^{n+m}g[j,\Gamma]\ /F^{*}_{p},\quad i=1,\ldots,n+m'\quad \Gamma=l+1,\ldots,l'$$

と生成できる。

【００４０】また、これらをまとめて

$$g''[i,\Gamma]=\Pi_{j=1}^{n+m}g[j,\Gamma]^{A[j,i]}\ /F^{*}_{p},\quad i=1,\ldots,n+m'\quad \Gamma=1,\ldots,l'$$

と記述できる。

【００４１】この時、$g''[i,\Gamma];i=1,\ldots,n+m;\ \Gamma=1,\ldots,l$を「出力文列」と呼ぶ。ここで、$g''[\mu,\Gamma]$が

表現値で、A[μ，ν]が表現で、g[μ，Γ]が基底である。

【００４２】変換情報保有コミットメントを応答の数に応じて、複数組生成する場合には、異なるA[μ，j]；μ＝1，...，n+m，j=n+1，...，n+m'を複数用意して生成する。

【００４３】この変換情報保有コミットメントと、入力文列と出力暗号文列および挑戦値に対応する応答を証明者が検証式を満たすように生成できることが、入力文列から出力暗号文列への変換の知識を有していることの証明となる。

【００４４】［変換条件コミットメント生成処理］証明付再暗号シャッフル方法を構成する変換条件コミットメント生成処理について説明する。

【００４５】応答と挑戦値の関係に、入力文列から出力文列および変換情報保有コミットメントへの変換の満たす条件が反映される。そのため、挑戦値に依存せずに成り立つ、応答と挑戦値の関係式が存在する。この関係式をこの変換の満たす条件を表現するものとしてコミットしたものを変換条件コミットメントとする。

【００４６】応答を複数個生成する場合は、知識の隠蔽因子の違いを関係式に反映させる。例えばこの関係式を、応答と挑戦値の多項式である恒等式とし、その係数をコミットするか、あるいはこの多項式の一部の項を準応答として、準応答の係数をコミットして変換条件コミットメントとできる。また挑戦値決定後に応答と準応答を生成すればよい。

【００４７】応答の各成分は挑戦値の多項式であるが、一部のこの多項式の、一部の項の各2乗の和と、挑戦値の成分の一部の各２乗の和とが、挑戦値によらずに等しくなる関係を内包しているものや、上記にて各３乗の和とが、挑戦値によらずに等しいくなる関係を内包している恒等式を実施例では用いている。

【００４８】これに対応する実施例における恒等式は、挑戦値c[i]、応答r[i]を用いて、

$$\Sigma_{i=1}^{n} r[i]r[i]r[i]+\rho''r'+\Sigma_{\mu=1}^{n+m}\rho'[\mu]r[\mu]/F_q$$
$$=\Sigma_{i=1}^{n} r[i]r[i]r[i]+\rho''(\lambda[0]+\Sigma_{i=1}^{n}\lambda[i]r[i]r[i])+\Sigma_{\mu=1}^{n+m}\rho'[\mu]r[\mu]/F_q$$
$$=\Sigma_{i=1}^{n} c[i]c[i]c[i]+\Sigma_{i=1}^{n}\psi[i]c[i]c[i]+\Sigma_{\mu=1}^{n+m}{}'\phi[\mu]c[\mu]/F_q$$

があげられる。

【００５４】ここでは、変換の満たす条件に対応する関係式を内包するように恒等式の係数ρ''，ρ'[i]，φ[μ]，ψ[i]を求めなければならない。

【００５５】また恒等式の一部

$r'=\lambda[0]+\Sigma_{i=1}^{n}\lambda[i]r[i]r[i]/F_q$

を準応答として、準元係数λ[μ]；μ=0，...，nをコミットする場合もある。

【００５６】変換条件コミットメントとして、恒等式の係数または、それをコミットしたものと、準元係数をコミットしたものを生成する。実施例では恒等式の一部を、

$v, v^{\lambda[\mu]}/F^*_p$

$*\ \Sigma_{i=1}^{n}(\Sigma_{j=1}^{n}A[i,j]c[j])^2=\Sigma_{i=1}^{n}c[i]^2/F_q$

や、

$\Sigma_{i=1}^{n}(\Sigma_{j=1}^{n}A[i,j]c[j])^3=\Sigma_{i=1}^{n}c[i]^3/F_q$

の関係を内包したものを用いている。

【００４９】なお、

$\Sigma_{j=1}^{n}A[i,j]c[j]/F_q \quad i=1,\ldots,n$

は、r[i]を構成する挑戦値の多項式

$\Sigma_{j=1}^{n+m}{}'A[i,j]c[j]/F_q \quad i=1,\ldots,n$

の一部である。

【００５０】例えばこれらの関係式は、入力文列から出力暗号文列および変換情報保有コミットメントへの変換を定義している変数A[μ，ν]；μ=0，...，n+m；ν=0，...，n+m'におけるA[i，j]；i，j=0，...，nが、正規直交行列であることや、準置換行列であることの性質を反映した関係式である。

【００５１】「置換行列」とは、正方行列で、どの行どの列にもただ一つだけ0でない成分が存在し、その値が１である行列のことである。正規直交行列でありかつ準置換行列である行列は置換行列である。

【００５２】「準置換行列」とは、上記置換行列の1である成分を、１の３乗根のいずれかで置き換えたものとする。ただし、それぞれの成分毎に異なる１の３乗根で置き換えを行っても良い。この時、置換行列に対応するの変換は再暗号シャッフルに対応している。すなわち、この変換条件コミットメント生成処理により変換の満たす条件を証明することによって変換が再暗号シャッフルであることを証明できる効果がある。

【００５３】例えば、上記関係式を内包した恒等式の例として、

$\Sigma_{i=1}^{n} r[i]r[i]+\Sigma_{\mu=1}^{n+m}\rho'[\mu]r[\mu]/F_q$
$=\Sigma_{i=1}^{n} c[i]c[i]+\Sigma_{\mu=1}^{n+m}{}'\phi[\mu]c[\mu]/F_q$

や、

$$\Sigma_{i=1}^{n} r[i]r[i]r[i]+\rho''r'+\Sigma_{\mu=1}^{n+m}\rho'[\mu]r[\mu]/F_q$$
$$=\Sigma_{i=1}^{n} r[i]r[i]r[i]+\rho''(\lambda[0]+\Sigma_{i=1}^{n}\lambda[i]r[i]r[i])+\Sigma_{\mu=1}^{n+m}\rho'[\mu]r[\mu]/F_q$$
$$=\Sigma_{i=1}^{n} c[i]c[i]c[i]+\Sigma_{i=1}^{n}\psi[i]c[i]c[i]+\Sigma_{\mu=1}^{n+m}{}'\phi[\mu]c[\mu]/F_q$$

のようにコミットし、準元係数を

$u, u^{\lambda[\mu]}/F^*_p \quad \mu=0,\ldots,n$

のようにコミットする。

【００５７】恒等式の係数をコミットすることと、準応答を用いることには、検証者が応答とコミットメントから再暗号シャッフルを特定するための情報を減じる効果がある。

【００５８】［応答生成処理］証明付再暗号シャッフル方法を構成する応答生成処理について説明する。

【００５９】応答生成処理では、変換情報保有コミットメントと変換条件コミットメントと入力文列と出力暗号文列を挑戦値生成関数に入力して挑戦値をえる。

【００６０】ここで、「挑戦値生成関数」とは、出力か

ら入力を求めることや、異る出力成分間の関係を意図して入力を決定することが計算量的に困難である関数である。これにより挑戦値が入力とコミットメントと出力とが決定後に、証明者の意図を入れずに生成されたことが保証できる。

【００６１】挑戦値生成関数を用いない場合には、検証者が、入力と出力とコミットメントとが示された後に無作為に選ぶことで挑戦値を得る。

【００６２】挑戦値から、再暗号シャッフル方法と情報隠蔽因子とを反映した応答や準応答を生成する。

【００６３】応答や準応答を複数個生成する場合は、各応答は異る情報隠蔽因子を反映させる。

【００６４】例えば、出力暗号文列および変換情報保有コミットメントを基底として挑戦値を表現とする表現値と、入力文列を基底として応答を表現とする表現値が等しくなるような応答を生成すればよい。

【００６５】例えば、応答$r[\mu];\mu=1,\ldots,n+m$は、挑戦値$c[\mu];\mu=1,\ldots,n+m'$を用いて、

$$r[\mu]=\Sigma_{\nu=1}^{n+m'}A[\mu,\nu]c[\nu]\ /F_q\quad \mu=1,\ldots,n+m$$

を、準応答として、

$$r'=\lambda[0]+\Sigma_{i=1}^{n}\lambda[i]r[i]r[i]\ /F_q$$

を生成する。

【００６６】［変換情報保有検証処理］再暗号シャッフル検証方法を構成する変換情報保有検証処理について説明する。

【００６７】入力文列と出力暗号文列と変換情報保有コミットメント間の関係を、応答と挑戦値の関係が反映していることを検証する。例えば、出力暗号文列および変換情報保有コミットメントを基底として挑戦値を表現とする表現値と、入力文列を基底として応答を表現とする＊表現値が等しくなるような応答と挑戦値の関係が存在することを確認する。

【００６８】例えば、挑戦値$c[i];i=1,\ldots,n+m'$と応答$r[i];i=1,\ldots,n+m$が、

$$\Pi_{i=1}^{n+m'}g''[i,\Gamma]^{c[i]}=\Pi_{i=1}^{n+m}g[i,\Gamma]^{r[i]}\ /F_p\quad \Gamma=1,\ldots,l'$$

が成り立つことを確認する。

【００６９】挑戦値は、証明文生成に用いた値と同じ値を用いる。これは、挑戦値生成関数を用いる場合は、挑戦値生成関数への入力が証明文と入出力に存在するので可能である。

【００７０】［変換条件検証処理］再暗号シャッフル検証方法を構成する変換条件検証処理について説明する。

【００７１】変換条件コミットメントより、挑戦値と応答が変換の満たす条件を反映した関係を満たしていることを検証する。

【００７２】例えば、変換の満たす条件を反映した関係を内包する恒等式に、応答と挑戦値または、応答と挑戦値と準応答を代入して、恒等式が成立することを確認している。また準応答がある場合、応答と準応答と準元係数をコミットしたものより、準応答の正当性も確認する。

【００７３】例えば、変換条件コミットメントとして恒等式の係数$\rho'',\rho'[\mu],\phi[\mu],\psi[i]$に対して、挑戦値$c[i];i=1,\ldots,n+m'$と、応答$r[i];i=1,\ldots,n+m$が、恒等式

$$\Sigma_{i=1}^{n}r[i]r[i]+\Sigma_{\mu=1}^{n+m}\rho'[\mu]r[\mu]=\Sigma_{i=1}^{n}c[i]c[i]+\Sigma_{\mu=1}^{n+m'}\phi[\mu]c[\mu]\ /F_q$$

や、恒等式

$$\begin{aligned}&\Sigma_{i=1}^{n}r[i]r[i]r[i]+\rho''r'+\Sigma_{\mu=1}^{n+m}\rho'[\mu]r[\mu]\ /F_q\\&=\Sigma_{i=1}^{n}r[i]r[i]r[i]+\rho''(\lambda[0]+\Sigma_{i=1}^{n}\lambda[i]r[i]r[i])+\Sigma_{\mu=1}^{n+m}\rho'[\mu]r[\mu]\ /F_q\\&=\Sigma_{i=1}^{n}c[i]c[i]c[i]+\Sigma_{i=1}^{n}\psi[i]c[i]c[i]+\Sigma_{\mu=1}^{n+m'}\phi[\mu]c[\mu]\ /F_q\end{aligned}$$

が成り立つことを確認し、また準応答の正当性を、検証式

$$u^{r'}=u[0]\Pi_{i=1}^{n}u[i]^{r[i]r[i]}\ /F_p$$

が成り立つことより確認する。

【００７４】また恒等式の係数の一部がコミットされて※いる場合は、代わりに、

$$\begin{aligned}&v^{\wedge}\{\Sigma_{i=1}^{n}r[i]r[i]r[i]+\rho''r'+\Sigma_{\mu=1}^{n+m}\rho'[\mu]r[\mu]\}\ /F_p\\&=v^{\wedge}\{\Sigma_{i=1}^{n}r[i]r[i]r[i]+\rho''(\lambda[0]+\Sigma_{i=1}^{n}\lambda[i]r[i]r[i])+\Sigma_{\mu=1}^{n+m}\rho'[\mu]r[\mu]\}\ /F_p\\&=v^{\wedge}\{\Sigma_{i=1}^{n}c[i]c[i]c[i]+\Sigma_{i=1}^{n}\psi[i]c[i]c[i]+\Sigma_{\mu=1}^{n+m'}\phi[\mu]c[\mu]\}\ /F_p\end{aligned}$$

や、

$$v^{\wedge}\{\Sigma_{i=1}^{n}r[i]r[i]+\Sigma_{\mu=1}^{n+m}\rho'[\mu]r[\mu]\}\ /F_p=v^{\wedge}\{\Sigma_{i=1}^{n}c[i]c[i]+\Sigma_{\mu=1}^{n+m'}\phi[\mu]c[\mu]\}\ /F_p$$

や、

が成り立つことを確認する。上式で、記号「^」は指数演算を示す。

【００７５】［入力文列生成方法］本発明に係る証明付再暗号シャッフル方法は、応答と挑戦値の関係に、入力文列から出力暗号文列と変換情報保有コミットメントへの変換を反映する必要があった。そのためには、挑戦値が与えられたときに生成できる応答が制限されていなければならない。しかし、入力暗号文の生成情報を証明者が知っている場合この制限を破れる可能性がある。そしてこれを阻止するための方法が入力文列生成方法である。

【００７６】本発明に係る入力文列生成方法は、疑似乱数を生成し、これにより入力文列を変換するか、この疑似乱数を入力文列に加えることにより、入力暗号文の生

成者にさえ入力文列を決定できない入力文列を生成する。

【００７７】［入力文列生成方法(1)］疑似乱数を生成して、入力暗号文列と公開鍵にくわえて入力文列とする。この時疑似乱数を決められた入力より決定して再現性を保証する。

【００７８】例えば、入力暗号文列を、$g[i,\Gamma];i=1,\ldots,n;\Gamma=0,\ldots,l$、公開鍵を$g[i,\Gamma];i=n+1,\ldots,n+m;\Gamma=0,\ldots,l$としたとき、決まった入力から疑似乱数を、$(n+m)\times(l'-l)$　$;l'-l\geqq 1$個生成し、これを、$g[i,\Gamma];i=1,\ldots,n+m;\Gamma=l+1,\ldots,l'$として、入力文列を、
$g[i,\Gamma]i=1,\ldots,n+m;\Gamma=1,\ldots,l'$
とする。

【００７９】［入力文列生成方法(2)］入力暗号文列を構成する各暗号文と公開鍵を、入力暗号文列と公開鍵を入力として生成した公開鍵列を構成する各公開鍵で再暗号化して、これをあわせて入力文列とする。

【００８０】ここで「公開鍵列」とは、入力から疑似乱数を公開鍵列を構成する公開鍵の数と同数一意的に生成し、この各乱数が各公開鍵のいずれかの要素となるようにしたものをいう。

【００８１】例えば、公開鍵列を、$g'[i,\Gamma];i=1,\ldots,n+m;\Gamma=1,\ldots,l$とし、入力暗号文列を、$\eta[i,\Gamma];i=1,\ldots,n;\Gamma=0,\ldots,l$、公開鍵を、$\eta[i,\Gamma];i=n+1,\ldots,n+m;\Gamma=0,\ldots,l$としたとき、入力文列　$g[i,\Gamma];i=1,\ldots,n+m;\Gamma=l+1,\ldots,l$を、検証者にも明らかな任意の正整数$s[i];i=1,\ldots,n+m$を用いて、
$g[i,\Gamma]=\eta[i,\Gamma]g'[i,\Gamma]^{s[i]}\quad /F^*_p$
と表す。$s[i]$として、例えば$s[n+m]=0, s[j]=1;j=1,\ldots,n+m-1$を選ぶ。

【００８２】［入力文列生成方法(3)］各入力平文を公開鍵列を構成する対応する各公開鍵を用いて暗号化し、この公開鍵を用いて暗号化したことの証明をする。この証明文が受理された暗号文と、公開鍵をあわせて入力文列とする。

【００８３】例えば、公開鍵列を、$g'[i,\Gamma];i=1,\ldots,n+m;\Gamma=0,1$とし、平文を、$m[i];i=1,\ldots,n;\Gamma=0,1$としたとき、入力暗号文を、
$\eta[i,0]=g'[i,0]^{s[i]}\quad /F^*_p,\quad i=1,\ldots,n$
$\eta[i,1]=m[i]g'[i,1]^{s[i]}\quad /F^*_p,\quad i=1,\ldots,n$
と生成し、合わせて、$\eta[i,0]=g'[i,0]^{s[i]}\quad /F^*_p$，　となる$s[i]$の知識を証明することで、$g'[i,0]$を使って暗号化した証明文とする。

【００８４】この証明文が検証された暗号文より、入力文列を
$g[i,\Gamma]=\eta[i,\Gamma]\quad i=1,\ldots,n;\Gamma=0,1$
$g[i,\Gamma]=g'[i,\Gamma]\quad i=n+1,\ldots,n+m;\Gamma=0,1$
とする。

【００８５】［証明付公開鍵列生成方法］与えられた入力から一意的に疑似乱数列を発生させ、その各乱数から与えられた手続きにより作られた値を成分に持ち、かつ同じ秘密鍵を持つ公開鍵を、各乱数に対応して複数個生成する。これと同時に、全ての公開鍵が同じ秘密鍵を持つことの証明文を生成する。

【００８６】複数人で秘密鍵を分散所持している場合は、各人が各秘密鍵で分散して公開鍵列を生成した後、それらを合わせて統合した公開鍵列を生成する。

【００８７】例えば、疑似乱数生成器Hash(*)が与えられ、入力*から出力を得る。そして出力をまた入力することを繰り返すことにより、再帰的に、疑似乱数列 を生成する。この数列の各値をk乗してできた数列から0,1を除いて得られるn+m個の値よりなる数列$g'[i,0];i=1,\ldots,n+m$の各値を成分に持ち、同じ秘密鍵を持つ公開鍵$g'[i,\Gamma];i=1,\ldots,n+m;\Gamma=0,\ldots,l$を各乱数に対応して生成する。

【００８８】ここで秘密鍵を$x[\Gamma];\Gamma=1,\ldots,l$とすると、公開鍵列は、
$g'[i,0]=g'[i,0]$
$g'[i,\Gamma]=g'[i,0]^{x[\Gamma]}\quad /F^*_p,\quad i=1,\ldots,n+m;\Gamma=1,\ldots,l$
と表せる。

【００８９】上記公開鍵列を正しく生成したことの証明文を生成する。

【００９０】秘密鍵を分散所持している場合、各自で分散秘密鍵に対応する公開鍵列を生成し、最後にそれらを合わせて秘密鍵に対する公開鍵列を生成する。

【００９１】
【実施例】本発明の実施例について図面を参照して説明する。以下の実施例では、Elgamal暗号を使った例に即して説明する。図中では、略語が使用されており、例えば保有コミットとは変換情報保有コミットメント、条件コミットとは変換条件コミットメント、恒等式コミットとは恒等式係数のコミットメント、準応答コミットとは準応答の係数のコミットメント、保有処理とは変換情報保有コミットメント生成処理、条件処理とは変換条件コミットメント生成処理、応答処理とは応答生成処理、保有検証処理とは変換情報保有検証処理、条件検証処理とは変換条件検証処理のことをいう。

【００９２】図３は、本発明に係る証明付再暗号シャッフル装置および再暗号シャッフル検証装置の実施例における入出力について示す図である。

【００９３】図３を参照すると、本発明の一実施例においては、複数の入力暗号文322と公開鍵323からなる入力文列300と、並び替え方を決めるシャッフル行列307と、再暗号化の変数である再暗号秘密乱数305と、変換情報保有コミットメントを生成するための乱数である情報隠蔽因子306と、からなる再暗号シャッフル行列304と、恒等式の係数の種となる元係数308と、恒等式の一部である準応答319の係数である準元係数309と、これらをコミットするための係数基底310とよりなる変換条件コミットメントを生成するための諸定数と、を含む再暗号シャ

ッフル情報303と、が、証明付再暗号シャッフル装置312に入力され、出力暗号文列313と再暗号シャッフル証明文314が出力される。

【００９４】再暗号シャッフル証明文314は、恒等式の係数、または、それをコミットしたものと準応答の係数をコミットしたものとを含む変換条件コミットメント316と、変換情報保有コミットメント315と、応答317と、準応答318とを含む。

【００９５】これら入力文列300と、出力暗号文列313と、再暗号シャッフル証明文314とが、再暗号シャッフル検証装置319に入力され、受理または不受理の検証結果320が出力される。

【００９６】上記証明付再暗号シャッフル装置は、証明者が入力暗号文列の生成情報を知らない時のみに、再暗号シャッフル証明文を偽造できない。この偽造をいかなる場合にも阻止するために加える方法が、入力文列生成方法であり、３種類の入力文列生成方法の実施例を挙げる。また、このうち二つの入力文列生成方法で使用される証明付公開鍵列生成方法についても説明する。

【００９７】以下では、証明付再暗号シャッフル方法と、入力文列生成方法と、証明付個別公開鍵列生成方法に共通して、前提となる事項から順に説明していく。

【００９８】［Elgamal領域変数］まずElgamal領域変数について説明する。

【００９９】この変数は、二つの素数p,qであり、これらは、

$p=kq+1$

なる関係を満たす。ここで、kは整数である。

【０１００】［挑戦値生成関数と基底生成関数］挑戦値生成関数と基底生成関数について説明する。これらは順に、

$Hash[\mu;\mu=0,\ldots,n](*), Hash'[\mu;\mu=0,\ldots,n](*)$

とする。

【０１０１】両関数の添字についているギリシャ文字μは0からnまでの値をとり、引数「*」を入力すると、それぞれn+1成分のベクトルを出力する。

【０１０２】挑戦値生成関数の出力は、n+1個の1,0でないq以下の整数、基底生成関数の出力は、n+1個の1,0でないp以下の整数で位数qのF^*_pの元(位数p-1の乗法群の位数q部分群の元)である整数である。

【０１０３】また、これらの関数は入出力間や出力の異なる成分間の関係を計算量的に意図して引数を決定できない関数とする。

【０１０４】基底生成関数の具体的な構成方法の例としては、|p|ビットを出力するハッシュ　関数Hash(*)を一つ用意して、

$Hash(*)$

を計算し、次に、この計算結果をハッシュ関数の引数に入力してさらに計算結果を得る。これを、繰り返すことにより、再帰的に、数列h[0],h[1],h[2],...を生成し、その各数値をk乗をした数列$h[0]^k, h[1]^k, h[2]^k, \ldots$を求める。この中から順に、1,0でないものをn+1個選んでいく。

【０１０５】挑戦値生成関数の場合は、|q|ビットを出力するハッシュ関数を用いて数列を求め、その中から順に1,0でないものをn+1個選んでいく(この場合、k乗する操作は必要無い)。

［公開鍵］公開鍵について説明する。公開鍵は、二つの数値$\eta[0,0]$, $\eta[0,1]$であり$\eta[0,0]$は位数qのF^*_pの元とする。$\eta[0,1]$は秘密鍵xを用いて、

$\eta[0,1]=\eta[0,0]^x \ /F^*_p$

と計算される。

【０１０６】［入力暗号文］入力暗号文について説明する。平文を、p以下で位数qのF^*_pの元から選び、これをMとする。これから疑似乱数生成器で生成した秘密乱数rを用いて、入力暗号文を、

$(\eta[0,0]^r, M\ \eta[0,1]^r)\ /F^*_p$

と計算する。

【０１０７】［再暗号化］再暗号化について説明する。Elgamal暗号文$(\eta[0,0]^r, M\ \eta[0,1]^r)\ /F^*_p$が与えられた時、任意の乱数sを選んで、

$(\eta[0,0]^r, M\eta[0,1]^r)\rightarrow(\eta[0,0]^r\ \eta[0,0]^s, M\eta[0,1]^r\ \eta[0,1]^s)\ /F^*_p=(\eta[0,0]^{r+s}, M\eta[0,1]^{r+s})\ /F^*_p$

なる変換を行うことを「再暗号化」という。上記変換はrを知らなくても実行できる。またこの変換により再暗号化された暗号文の復号結果はかわらない。この時の乱数sを、「再暗号秘密乱数」と呼ぶ。

【０１０８】［置換行列］置換行列について説明する。「置換行列」とはどの行にもまたどの列に対しても０でない成分が唯一存在し、１の値をとる。ただし本実施例ではFq上で考える。下に例をあげる。

【０１０９】

$$\begin{matrix}0&0&0&1&0\\1&0&0&0&0\\0&1&0&0&0\\0&0&0&0&1\\0&0&1&0&0\end{matrix}\ /F_q$$

【０１１０】［準置換行列］準置換行列ついて説明する。「準置換行列」とは、置換行列の1である成分を、F^*_p上の３個ある１の３乗根のいずれかで置き換えたものと定義する。これらをw,w^2,1として下に準置換行列の例をあげる。

【０１１１】

$$\begin{matrix}0&0&0&w^2&0\\w&0&0&0&0\\0&w^2&0&0&0\\0&0&0&0&1\\0&0&w&0&0\end{matrix}\ /F_q$$

【０１１２】［再暗号シャッフル］再暗号シャッフルについて説明する。入力暗号文列$\eta[i,0]$, $\eta[i,1]$;i=

1,...,nの順序を入れ替えて、暗号文列η'[i,0],η'[i,1];i=1,...,nを生成し、さらにn個の秘密乱数s[i]:i=1,...,nと、公開鍵η[0,0],η[0,1]を用いて、出力暗号文列g''[i,Γ];i=1,...,n,Γ=0,1を、

$g''[i,\Gamma]=\eta'[i,\Gamma]\,\eta[0,\Gamma]^{s[i]}\quad /F^{*}_{p}\quad i=1,\ldots,n,\ \Gamma=0,1$

と計算する。これが、再暗号シャッフルの出力結果である。これを、「出力暗号文列」と呼ぶ。

【０１１３】［再暗号シャッフル行列］再暗号シャッフル行列について説明する。「再暗号シャッフル行列」とは、n+1行n+1列の行列で、その成分A[μ,ν];μ,ν=0,...,nが、

A[μ,ν]=

A[i,j]　i,j=1,...,n　シャッフル行列307

$A[0,j]\in_{R}$　j=1,...,n　再暗号秘密乱数305

$A[i,0]\in_{R}$　i=1,...,n　情報隠蔽因子306

$A[0,0]\in_{R}$　情報隠蔽因子306

であるものである。

【０１１４】［再暗号シャッフル行列変換］再暗号シャッフル行列変換について説明する。これは、入力文列g[μ,Γ]に以下のように作用して、出力文列g''[μ,Γ]を出力する。

【０１１５】$g''[\mu,\Gamma]=\Pi_{\nu=0}^{n} g[\nu,\Gamma]^{A[\nu,\mu]}\ /F^{*}_{p}\quad \mu=0,\ldots,n,\ \Gamma=0,1$

ここで、シャッフル行列が置換行列の場合、出力暗号文列を、g''[i,0],g''[i,1];i=1,...,nとし、展開すると、ある置換(i,j|π(i)=j)に対して、

$g''[j,0]=g[i,0]\,\eta[0,0]^{A[0,j]}\quad /F^{*}_{p}$

$g''[j,1]=g[i,1]\,\eta[0,1]^{A[0,j]}\quad /F^{*}_{p}$

となり、これは再暗号シャッフルの出力となる。

【０１１６】またシャッフル行列が準置換行列の場合、準再暗号シャッフルの結果

$g''[j,0]=g[i,0]^{w[i]}\quad \eta[0,0]^{A[0,j]}\quad /F^{*}_{p}$

$g''[j,1]=g[i,1]^{w[i]}\quad \eta[0,1]^{A[0,j]}\quad /F^{*}_{p}$

を出力する(準再暗号シャッフルとは各出力暗号文を1またはwまたはw^{2}乗すると再暗号シャッフルとなるものと定義する)。ここでw[i];i=1,...,nはFq上の1の三乗根のいずれかをとる。

【０１１７】［実施例(1)］本発明の一実施例をなす証明付再暗号シャッフル方法およびその検証方法について、図４、図５を参照して説明する。以下で、Γ=0,1を取るものとする。

【０１１８】再暗号シャッフル情報401として、再暗号シャッフル行列402、係数基底404、元係数403を以下のように準備する。

【０１１９】再暗号シャッフル行列402に関しては、まず１からnまでの数を順に並べる。疑似乱数発生器（不図示）をn回使って、n個数列を発生させ、そのi番目の数をn-i+1で割り余りの数を求めπ'(i)とする。

【０１２０】iは1からn迄順に、上記並べた数の下からπ'(i)番目の数をπ(i)とし、上記数列からこの数を取り除く作業を行いπ(i);i=1,...,nを決定する。シャッフル行列の第i行目はπ(i)列目の成分のみ値を1とし、その他を0とする。以上のようにして置換行列を生成する。

【０１２１】再暗号シャッフル行列のシャッフル行列以外の成分を以下のようにして生成する。まず、疑似乱数発生器により2n+1個のF_{q}上の数を作成し、A[i,0],A[0,j],A[0,0];i,j=1,...,nに割り振る。以上を合わせて再暗号シャッフル行列とする。

【０１２２】係数基底404　v、元係数403 r'[0]を生成に関しては、疑似乱数生成器で1,0でないF_{q}上の数を生成し、r'[0]とし、疑似乱数生成器により$/F^{*}_{p}$の元を生成しF^{*}_{p}上でそのk乗をとり1,0でないものを選び位数qのF^{*}_{p}の元を生成し、vとする。

【０１２３】$r'[0]\in_{R}\ F_{q}\ ,\neq 0,1$

$v\in_{R}\ F^{*}_{p}\ ,\neq 1\ ,\text{s.t.}\ v^{q}=1\ /F^{*}_{p}$

入力暗号文列η[i,0],η[i,1];i=1,...,nと、公開鍵η[0,0],η[0,1]より、入力文列400g[μΓ];μ=0,...,n;Γ=0,1を、

$g[0,\Gamma]=\eta[0,\Gamma]\quad \Gamma=0,1$

$g[i,\Gamma]=\eta[i,\Gamma]\ /F^{*}_{p}\quad i=1,\ldots,n,\ \Gamma=0,1$

とする。

【０１２４】以下、証明付再暗号シャッフル方法を使う。

【０１２５】変換情報保有コミットメント生成処理419における再暗号シャッフル行列作用405により、上記暗号シャッフル行列402を入力文列400に以下の様に作用させて、出力文列406　g''[μ,Γ];μ=0,...,n;Γ=0,1を、

$g''[\mu,\Gamma]=\Pi_{\nu=0}^{n} g[\nu,\Gamma]^{A[\nu,\mu]}\ /F^{*}_{p}\quad \mu=0,\ldots,n,\ \Gamma=0,1$

と生成する。

【０１２６】ここで、g''[i,Γ];i=1,...,n;Γ=0,1を出力暗号文列407、g''[0,Γ];Γ=0,1を変換情報保有コミットメント408とする。

【０１２７】変換条件コミットメント生成処理420における恒等式係数計算409により、元係数403　r'[0]と、再暗号シャッフル行列402と、を用いて、恒等式係数410　φ[μ],r'[0]を、

r'[0]=r'[0]

$\phi[0]=\Sigma_{j=1}^{n} A[j,0]A[j,0]+r'[0]A[0,0]\ /F_{q}$

$\phi[i]=2\Sigma_{j=1}^{n} A[j,0]A[j,i]+r'[0]A[0,i]\ /F_{q}\quad i=1,\ldots,n$

と生成する。

【０１２８】さらに、係数基底404　vを用いて、隠蔽処理411により、恒等式係数410　r'[0],φ[0]を、

$v'=v^{r'[0]}\quad /F^{*}_{p}$

$\omega=v^{\phi[0]}\quad /F^{*}_{p}$

とコミットする。

【０１２９】以上より、$\phi[i], \omega, v', v$を変換条件コミットメント412とする。

【０１３０】ここで、コミットメント40 Aを、変換情報保有コミットメント408と変換条件コミットメント412とする。

【０１３１】応答生成処理421により、以上の入力文列400と、出力暗号文列417と、コミットメント409と、を挑戦値生成関数413の引数として、挑戦値414を、

$c[0]=1,$

$c[i]=Hash[i](g[\nu,0], g[\nu,1], g''[\nu,0], g''[\nu,1], v, \phi[\nu], \omega, v'; \nu=0,\ldots,n)\quad i=1,\ldots,n$

と生成し、この挑戦値から再暗号シャッフル行列02を用いて、応答416を$r[\mu]=\Sigma_{\nu=0}^{n} A[\mu,\nu]c[\nu]\ /F_q\quad \mu=0,\ldots,n$と生成415する。

【０１３２】以上のコミットメント40 Aと応答416を、再暗号シャッフル証明文418として出力し、再暗号シャッフルの結果として出力暗号文列417を出力する。

【０１３３】検証方法について、図５を参照して説明する。

【０１３４】再暗号シャッフル検証方法により、入力文列400　$g[\mu,\Gamma];\mu=0,\ldots,n;\Gamma=0,1$と、出力暗号文列417　$g''[i,\Gamma];i=1,\ldots,n;\Gamma=0,1$と、再暗号シャッフル証明文418中のコミットメント409である変換情報保有コミットメント408　$g''[0,\Gamma];\Gamma=0,1$と、変換条件コミットメント412　$\phi[\nu], \omega, v', v; \nu=0,\ldots,n;\Gamma=0,1$と、を挑戦値生成関数500に代入して、挑戦値501を、

$c[0]=1$

$c[i]=Hash[i](g[\nu,\Gamma], g''[\nu,\Gamma], \phi[\nu], \omega, v', v; \nu=0,\ldots,n;\Gamma=0,1)i=1,\ldots,n$

と生成する。

【０１３５】変換情報保有検証処理505により、この挑戦値501を用いて入力文列400と、変換情報保有コミットメント408と出力暗号文列417である出力文列406と応答416とを用いて検証式、

$\Pi_{\mu=0}^{n} g[\mu,\Gamma]^{r[\mu]}=\Pi_{\mu=0}^{n} g''[\mu,\Gamma]^{c[\mu]}\ /F^*_p\quad \Gamma=0,1$

が成り立つことを確認502する。

【０１３６】変換条件検証処理506により、挑戦値501と応答416と変換条件コミットメント412とを用いて検証式

$v'^{r[0]}\ \hat{}\ \{\Sigma_{i=1}^{n} r[i]r[i]\}=\omega\hat{}\{\Sigma_{i=1}^{n}(c[i]c[i]+\phi[i]c[i])\}\ /F^*_p$

が成り立つことを確認503する。

【０１３７】以上全ての検証式が成り立てば、証明文を受理504する。

【０１３８】上記証明付再暗号シャッフル方法は、入力文列に対する再暗号シャッフル行列変換が少なくとも正規直交行列に属するシャッフル行列を持つ再暗号シャッフル行列により行われたことを保証する効果がある。

【０１３９】入力暗号文と出力暗号文に制限が課せられており、この効果で、再暗号シャッフルの正当性を保証できる場合には、本実施例により、証明付再暗号シャッフルを構成できる。

【０１４０】例えば、入力暗号文は限られた候補から選ばれていることが証明されていて、かつそれらの候補は互いに他を基底として表現できないとする。この入力暗号文を再暗号シャッフルしたのち、復号していずれの復号文も正しい候補から選ばれたものであったとき、本実施例による証明文からこの再暗号シャッフルが正当であることが言える。

【０１４１】なお図４における変換情報保有コミットメント処理419、変換条件コミットメント生成処理420、応答生成処理421は、コンピュータ上で実行されるプログラムによりその処理・機能が実現される。また図５における変換情報保有検証処理505、変換条件検証処理506は、コンピュータ上で実行されるプログラムによりその処理・機能が実現される。この場合、該プログラムを記録した記録媒体（例えばＣＤ－ＲＯＭ、ＤＶＤ（digital versatile disk）、フロッピディスク媒体、ハードディスク媒体、磁気テープ媒体、半導体メモリ等のいずれか）から、該プログラムをコンピュータの主記憶にロードして実行することで、本発明を実施することができる。

【０１４２】［実施例(2)］本発明の実施例(2)の証明付再暗号シャッフル方法およびその検証方法について、図６、図７を参照して説明する。以下では、$\Gamma=0,1$を取るものとする。

【０１４３】再暗号シャッフル情報601として、再暗号シャッフル行列602、元係数603、係数基底604、605、準元係数606を、以下のように準備する。

【０１４４】まず再暗号シャッフル行列602に関しては、前記実施例(1)と同様に生成する。

【０１４５】元係数603　ρ', ρ''、係数基底604　v、係数基底605　u、準元係数606　$\lambda[\mu];\mu=0,\ldots,n$に関しても、実施例(1)と同様な手法で、$\rho', \rho''\ \lambda[\mu];\mu=0,\ldots,n$には1,0でない$F_q$上の数を、係数基底u,vには位数qの$F_p$の元を生成する。

【０１４６】$\rho'\in_R F_q\ ,\neq 0,1$

$\rho''\in_R F_q\ ,\neq 0,1$

$v\in_R F^*_p\ ,\neq 1,\ s.t.\ v^q=1\ /F^*_p$

$\lambda[\mu]\in_R F_q\ ,\neq 0,1,\ \mu=0,\ldots,n$

$u\in_R F^*_p\neq 1,\ s.t\ .u^q=1\ /F^*_p$

【０１４７】実施例(1)と同様にして、入力暗号文列と公開鍵より、入力文列600　$g[\mu\Gamma];\mu=0,\ldots,n;\Gamma=0,1$を生成する。

【０１４８】以下、証明付再暗号シャッフル方法を用いる。

【０１４９】実施例(1)と同様に、変換情報保有コミットメント生成処理623を行い、出力文列603　$g''[\mu,\Gamma];\mu=0,\ldots,n;\Gamma=0,1$を、生成する。ここで、$g''[i,\Gamma];i=1,\ldots,n;\Gamma=0,1$を、出力暗号文列604、g''[0,

Γ]；Γ=0,1を変換情報保有コミットメント605とする。

【０１５０】変換条件コミットメント生成処理625における恒等式係数計算606により、元係数603　ρ'，ρ''，と、再暗号シャッフル行列602を用いて恒等式係数607　ψ[i]，φ[i]，φ[0]，ρ'，ρ''；i=1,...,nを、

ρ'=ρ'

ρ''=ρ''

ψ[i]=$\Sigma_{j=1}^{n}$(3A[j,0]+ρ''λ[j])A[j,i] /F_q　i=1,...,n

φ[i]=$\Sigma_{j=1}^{n}$(3A[j,0]A[j,0]A[j,i]+2ρ''λ[j]A[j,0]A[j,i])+ρ'A[0,i] /F_q i=1,...,n

φ[0]=$\Sigma_{j=1}^{n}$(A[j,0]A[j,0]A[j,0]+ρ''λ[j]A[j,0]A[j,0])+ρ''λ[0]+ρ'A[0,0] /F_q

と生成する。

【０１５１】さらに、係数基底604　vを用いて、隠蔽処理608により、恒等式係数607　ρ'，ρ''，φ[0]を、

$\omega=v^{\phi[0]}$ /F^*_p

$v''=v^{\rho''}$ /F^*_p

$v'=v^{\rho'}$ /F^*_p

とコミット609する。さらに、係数基底605　uを用いて、準元係数606　λ[μ]；μ=0,...,nを、

$u[0]=u^{\lambda[0]}$ /F^*_p

$u[i]=u^{\lambda[i]}$ /F^*_p　i=1,...,n

とコミット612する。

【０１５２】以上より、ψ[i]，φ[i]，ω，v''，v'，v，u，u[0]，u[i]；i=1,...,nを変換条件コミットメント613とする。

【０１５３】ここで、コミットメント614を、変換情報保有コミットメント605と、変換条件コミットメント613とする。

【０１５４】応答生成処理624により、以上の入力文列600と、出力暗号文列604と、コミットメント614と、を挑戦値生成関数615の引数として、挑戦値616を、

c[0]=1

c[i]=Hash[i](g[ν，Γ]，g''[ν，Γ]，u，u[ν]，v，φ[j]，ψ[j]，ω，v'，v''；Γ=0,1,2；ν=0,...,n；j=1,...,n)　i=1,...,n

と生成し、この挑戦値616から、再暗号シャッフル行列602を用いて、応答618を、

r[μ]=$\Sigma_{\nu=0}^{n}$A[μ，ν]c[ν] /F_q　μ=0,...,n

と生成617する。

【０１５５】さらに、準応答620を、準元係数606　λ[μ]；μ=0,...,nと、応答618より、

r'=λ[0]+$\Sigma_{i=1}^{n}$λ[i]r[i]r[i] /F_q

と生成619する。

【０１５６】以上のコミットメント614と、応答618と、準応答620と、を再暗号シャッフル証明文622として出力し、再暗号シャッフルの結果として、出力暗号文列604を出力する。　検証方法について、図６及び図７を参照して、以下に説明する。

【０１５７】再暗号シャッフル検証方法により、入力文列600　g[μ，Γ]；μ=0,...,n；Γ=0,1と、出力暗号文列604　g''[i，Γ]；i=1,...,n；Γ=0,1と、再暗号シャッフル証明文622中のコミットメント614である変換情報保有コミットメント605　g''[0，Γ]；Γ=0,1と、変換条件コミットメント609、912　ψ[i]，φ[i]，ω，v''，v'，v，u，u[0]，u[i]；i=1,...,nと、を挑戦値生成関数704に代入して挑戦値705を、

c[0]=1

c[i]=Hash[i](g[ν，Γ]，g''[ν，Γ]，u，u[ν]，v，φ[jj，ψ[j]，ω，v'，v''；Γ=0,1,2；ν=0,...,n；j=1,...,n)i=1,...,n

と生成する。

【０１５８】変換情報保有検証処理710により、この挑戦値705を用いて、入力文列600と、変換情報保有コミットメント605と、出力暗号文列604である出力文列603と、応答618と、を用いて検証式、

$\Pi_{\mu=0}^{n}g[\mu,\Gamma]^{r[\mu]}=\Pi_{\mu=0}^{n}g''[\mu,\Gamma]^{c[\mu]}$ /F^*_p

Γ=0,1

が成り立つことを確認706する。

【０１５９】変換条件検証処理711により、挑戦値705と、応答618と、変換条件コミットメント609、612と、を用いて検証式、

$v''^{r'}$ $v'^{r[0]}$ $v^{\Sigma_{i=1}^{n}r[i]r[i]r[i]}=\omega v^{\Sigma_{i=1}^{n}(c[i]c[i]c[i]+\psi[i]c[i]c[i]+\phi[i]c[i])}$ /F^*_p

と検証式707

$u^{r'}=\Pi_{i=1}^{n}u[i]^{r[i]r[i]}$ /F^*_p

が成り立つこと確認708する。

【０１６０】以上全ての検証式が成り立てば証明文を受理709する。

【０１６１】上記証明付再暗号シャッフル方法は、入力文列に対する再暗号シャッフル行列変換が少なくとも置換行列に属するシャッフル行列を持つ再暗号シャッフル行列により行われたことを保証する効果がある。この時、出力暗号文列g''[iΓ]；i=1,...,n；Γ=0,1は、

$g''[j,0]=g[i,0]^{w[i]}$ $g[0,0]^{A[0,j]}$ /F^*_p

$g''[j,1]=g[i,1]^{w[i]}$ $g[0,1]^{A[0,j]}$ /F^*_p

を出力した可能性を排除できない。ここで、w[i]が全て1の時が、再暗号シャッフルである。なお、w[i]；i=1,...,nは、F_q上の1の三乗根のいずれかをとる。

【０１６２】そこで、復号文として、F_q上の1の三乗根乗の自由度を許すか、平文に決められた記号を記して三乗根乗の自由度を消せば、本実施例をもって、証明付再暗号シャッフルを構成できる。

【０１６３】なお証明付再暗号シャッフル装置の変換情報保有コミットメント処理623、変換条件コミットメント生成処理625、応答生成処理624は、コンピュータ上で実行されるプログラムによりその処理・機能が実現される。また証明付再暗号シャッフル検証装置の変換情報保有検証処理710、変換条件検証処理711は、コンピュータ

上で実行されるプログラムによりその処理・機能が実現される。この場合、該プログラムを記録した記録媒体（例えばＣＤ－ＲＯＭ、ＤＶＤ（digital versatile disk）、フロッピディスク媒体、ハードディスク媒体、磁気テープ媒体、半導体メモリ等のいずれか）から、該プログラムをコンピュータの主記憶にロードして実行することで、本発明を実施することができる。

【０１６４】［実施例(3)］本発明の実施例(3)として証明付再暗号シャッフル方法およびその検証方法について、図８、図９を参照して以下に説明する。以下で、$\Gamma=0,1$を取るものとする。また公開鍵は、$\eta[-1,\Gamma]$, $\eta[0,\Gamma];\Gamma=0,1$の二組あり、どちらも同じ秘密鍵を持つとする。

【０１６５】再暗号シャッフル情報801として、再暗号シャッフル行列802、元係数803,805、係数基底804,806、準元係数807を以下のように準備する。

【０１６６】本実施例で用いる再暗号シャッフル行列802は、実施例(1)と実施例(2)のものとは大きさが異り、n+2行n+1列の行列である。

【０１６７】この再暗号シャッフル行列802を構成するシャッフル行列は、$A[i,j];i,j=1,\ldots,n$であり、再暗号秘密乱数は、$A[-1j],A[0,j];j=1,\ldots,n$の$2\times n$成分であり、知識の隠蔽因子は、$A[\mu,0];\mu=-1,\ldots,n$のn+2成分である。これらの成分を、実施例(1)と同様に生成する。

【０１６８】元係数803　$r'[-1],r'[0]$、元係数805　ρ,ρ',ρ''、係数基底804　v、係数基底806　u、準元係数807　$\lambda[\mu];\mu=0,\ldots,n$に関しても、実施例(1)と同様な手法で、$r'[-1],r'[0],\rho,\rho',\rho'',\lambda[\mu];\mu=0,\ldots,n$には、1,0でない$F_q$上の数を、係数基底u,vには、位数qの$F^*_p$の元を生成する。

【０１６９】
$r'[-1]\in_R F_q,\neq 0,1$
$r'[0],\in_R F_q,\neq 0,1$
$\rho\in_R F_q,\neq 0,1$
$\rho'\in_R F_q,\neq 0,1$
$\rho''\in_R F_q,\neq 0,1$
$v\in_R F^*_p,\neq 0,1,\text{s.t.}\quad v^q=1\ /F^*_p$
$\lambda[\mu]\in_R F_q,\neq 0,1\ \mu=0,\ldots,n$
$u\in_R F^*_p,\neq 0,1,\text{s.t.}\quad u^q=1\ /F^*_p$

【０１７０】入力暗号文列　$\eta[i,0],\eta[i,1];i=1,\ldots,n$と、公開鍵$\eta[-1,\Gamma],\eta[0,\Gamma];\Gamma=0,1$より、入力文列800　$g[\mu\Gamma];\mu=-1,\ldots,n;\Gamma=0,1$を、

$g[-1,\Gamma]=\eta[-1,\Gamma]\quad\Gamma=0,1$
$g[0,\Gamma]=\eta[0,\Gamma]\quad\Gamma=0,1$
$g[i,\Gamma]=\eta[i,\Gamma]\ /F^*_p\quad i=1,\ldots,n,\Gamma=0,1$

とする。

【０１７１】以下証明付再暗号シャッフル方法を用いる。

【０１７２】変換情報保有コミットメント生成処理832における再暗号シャッフル行列作用808により、上記再暗号シャッフル行列802を入力文列800に以下の様に作用させて、出力文列809$g''[\mu,\Gamma];\mu=0,\ldots,n;\Gamma=0,1$を、

$g''[\mu,\Gamma]=\Pi_{\nu=1}{}^{n} g[\nu,\Gamma]^{A[\nu,\mu]}\ /F^*_p\quad\mu=0,\ldots,n,\Gamma=0,1$

と生成する。ここで、$g''[i,\Gamma];i=1,\ldots,n;\Gamma=0,1$を出力暗号文列810、$g''[0,\Gamma];\Gamma=0,1$を変換情報保有コミットメント811とする。

【０１７３】変換条件コミットメント生成処理833、834における恒等式係数計算812、816により、元係数803、805　$r'[-1]$、$r'[0],\rho,\rho',\rho''$,と、再暗号シャッフル行列802を用いて、恒等式係数817　$\psi[i],\phi[i],\phi[0],\rho,\rho',\rho'';i=1,\ldots,n$　と、恒等式係数813　$\Phi[\nu],r'[0],r'[-1];\nu=0,\ldots,n$　を計算818、812する。

【０１７４】
$\rho=\rho$
$\rho'=\rho'$
$\rho''=\rho''$
$\psi[i]=\Sigma_{j=1}{}^{n}(3A[j,0]+\rho''\lambda[j])A[j,i]\ /F_q\quad i=1,\ldots,n$
$\phi[i]=\Sigma_{j=1}{}^{n}(3A[j,0]A[j,0]A[j,i]+2\rho''\lambda[j]A[j,0]A[j,i])+\rho'A[0,i]+\rho A[-1,i]\ /F_q\quad i=1,\ldots,n$
$\phi[0]=\Sigma_{j=1}{}^{n}(A[j,0]A[j,0]A[j,0]+\rho''\lambda[j]A[j,0]A[j,0])+\rho''\lambda[0]+\rho'A[0,0]+\rho A[-1,0]\ /F_q$
$r'[-1]=r'[-1]$
$r'[0]=r'[0]$
$\Phi[0]=\Sigma_{j=1}{}^{n}A[j,0]A[j,0]+r'[0]A[0,0]+r'[-1]A[-1,0]\ /F_q$
$\Phi[i]=2\Sigma_{j=1}{}^{n}A[j,0]A[j,i]+r'[0]A[0,i]+r'[-1]A[-1,i]\ /F_q\quad i=1,\ldots,n$

【０１７５】さらに係数基底804　vを用いて、隠蔽処理814、818により、恒等式係数813,817　$r'[-1],r'[0],\Phi[0],\phi[0],\rho,\rho',\rho''$を、

$\omega=v^{\phi[0]}\quad/F^*_p$
$v''=v^{\rho\cdot\cdot}\quad/F^*_p$
$v'=v^{\rho\cdot}\quad/F^*_p$
$\omega'=v^{\rho}\quad/F^*_p$

とコミット819し、

$V=v^{r'[-1]}\quad/F^*_p$
$V'=v^{r'[0]}\quad/F^*_p$
$\Omega=v^{\Phi[0]}\quad/F^*_p$

とコミット815する。

【０１７６】さらに、係数基底806 uを用いて、準元係数807　$\lambda[\mu];\mu=0,\ldots,n$を、

$u[0]=u^{\lambda[0]}\quad/F^*_p$
$u[i]=u^{\lambda[i]}\quad/F^*_p\quad i=1,\ldots,n$

とコミット821、820する。

【０１７７】以上より、$\Phi[i],V',V,\Omega,\psi[i],\phi[i],\omega,v'',v',\omega',v,u,u[0],u[i];i=1,\ldots,n$を変換条件コミットメント822とする。

【０１７８】ここで、コミットメント823を，変換情報

保有コミットメント811と変換条件コミットメント822とする。

【０１７９】応答生成処理835により、以上の入力文列800と、出力暗号文列810と、コミットメント823を、挑戦値生成関数824の引数として、挑戦値825を、

$c[0]=1$

$c[i]=Hash[i](g[\mu,\Gamma],g''[\nu,\Gamma],u[\nu],u,\phi[j],\psi[j],\omega,\omega',v',v'',v,\Phi[j],\Omega,V',V;\mu=-1,\ldots,n;\nu=0,\ldots,n;j=1,\ldots,n;\Gamma=0,1,2)i=1,\ldots,n$

と生成し、この挑戦値825から、再暗号シャッフル行列802を用いて、応答827を、

$r[\mu]=\Sigma_{\nu=0}^{n}A[\mu,\nu]c[\nu]\ /F_q \quad \mu=-1,\ldots,n$

と生成826する。

【０１８０】さらに、準応答829を、準元係数807 $\lambda[\mu];\mu=0,\ldots,n$と応答827より、

$r'=\lambda[0]+\Sigma_{i=1}^{n}\lambda[i]r[i]r[i]\ /F_q$

と生成828する。

【０１８１】以上のコミットメント823と、応答827と、準応答829とを再暗号シャッフル証明文831として出力し、再暗号シャッフルの結果として、出力暗号文列810を出力する。

【０１８２】検証方法について図９を参照して説明する。

【０１８３】再暗号シャッフル検証方法により、入力文列800 $g[\mu,\Gamma];\mu=-1,\ldots,n;\Gamma=0,1$と、出力暗号文列810 $g''[i,\Gamma];i=1,\ldots,n;\Gamma=0,1$と、再暗号シャッフル証明文831中コミットメント823の変換情報保有コミットメント811 $g''[0,\Gamma];\Gamma=0,1$と、変換条件コミットメント815、819、821 $\Phi[i],V',V,\Omega,\psi[i],\phi[i],\omega,v'',v',\omega',v,u,u[0],u[i];i=1,\ldots,n$と、を、挑戦値生成関数900に代入して、挑戦値901を、

$c[0]=1$

$c[i]=Hash[i](g[\mu,\Gamma],g''[\nu,\Gamma],u[\nu],u,\phi[j],\psi[j],\omega,\omega',v',v'',v,\Phi[j],\Omega,V',V;\mu=-1,\ldots,n;\nu=0,\ldots,n;j=1,\ldots,n;\Gamma=0,1,2)i=1,\ldots,n$

と生成する。

【０１８４】変換情報保有検証処理907により、この挑戦値901を用いて入力文列800と、変換情報保有コミットメント811と、出力暗号文列810である出力文列809と、応答827とを用いて検証式、

$\Pi_{\mu=1}^{n}g[\mu,\Gamma]^{r[\mu]}=\Pi_{\mu=0}^{n}g''[\mu,\Gamma]^{c[\mu]}\ /F_p \quad \Gamma=0,1$

が成り立つことを確認902する。

【０１８５】変換条件検証処理908、909により、挑戦値901と応答827と準応答829と変換条件コミットメント815、819、821とを用いて、検証式

$v''^{r'}\ v'^{r[0]}\ \omega'^{r[-1]}\ v^{\wedge}\{\Sigma_{i=1}^{n}r[i]r[i]r[i]\}=\omega v^{\wedge}\{\Sigma_{i=1}^{n}(c[i]c[i]c[i]+\psi[i]c[i]c[i]+\phi[i]c[i])\}\ /F_p$

が成り立つことを確認904し、検証式

$u^{r'}=u[0]\Pi_{i=1}^{n}u[i]^{r[i]r[i]}\ /F_p$

が成り立つことを確認905し、検証式

$V'^{r[0]}\ V^{r[-1]}\ v^{\wedge}\{\Sigma_{i=1}^{n}r[i]r[i]\}=\Omega v^{\wedge}\{\Sigma_{i=1}^{n}(c[i]c[i]+\Phi[i]c[i])\}\ /F_p$

が成り立つこと確認903する。

【０１８６】以上全ての検証式が成り立てば証明文を受理906する。

【０１８７】上記証明付再暗号シャッフル方法は、入力文列に対する再暗号シャッフル行列変換が少なくとも置換行列に属するシャッフル行列を持つ再暗号シャッフル行列により行われたことを保証する効果がある。これは再暗号シャッフルが行われたことを意味し、本実施例は証明付再暗号シャッフルである。

【０１８８】なお証明付再暗号シャッフル装置の変換情報保有コミットメント処理832、変換条件コミットメント生成処理833、834、応答生成処理835は、コンピュータ上で実行されるプログラムによりその処理・機能が実現される。また証明付再暗号シャッフル検証装置の変換情報保有検証処理907、変換条件検証処理908、909は、コンピュータ上で実行されるプログラムによりその処理・機能が実現される。この場合、該プログラムを記録した記録媒体（例えばＣＤ－ＲＯＭ、ＤＶＤ（digital versatile disk）、フロッピディスク媒体、ハードディスク媒体、磁気テープ媒体、半導体メモリ等のいずれか）から、該プログラムをコンピュータの主記憶にロードして実行することで、本発明を実施することができる。

【０１８９】［実施例(4)］本発明の実施例(4)の証明付再暗号シャッフル方法およびその検証方法について図１０、図１１を参照して以下に説明する。以下で、$\Gamma=0,1$を取るものとする。また公開鍵は、$\eta[0,\Gamma];\Gamma=0,1$の一組である。

【０１９０】再暗号シャッフル情報1006として、再暗号シャッフル行列1001と、第２の情報隠蔽因子1004と、元係数1002、1005と、係数基底1003、1008と、準元係数1007を以下のように準備する。

【０１９１】まず再暗号シャッフル行列1001に関しては、前記実施例(1)と同様に生成し、これを$\underline{A}[\mu,\nu];\mu,\nu=0,\ldots,n$とする。

【０１９２】さらに、第２の情報隠蔽因子1004 $A[\nu,0];\nu=0,\ldots,n$を同様に生成する。

【０１９３】元係数1005 ρ',ρ''、元係数1002 $r'[0]$、係数基底1003 v、係数基底1008u、準元係数1007 $\lambda[\mu];\mu=0,\ldots,n$ に関しても、実施例(1)と同様な手法で、$r'[0],\rho',\rho'',\lambda[\mu];\mu=0,\ldots,n$には1,0でない$F_q$上の数を、係数基底u,vには位数qの$F_p$の元を生成する。

【０１９４】$\rho'\in_R F_q\ ,\neq 0,1$

$\rho''\in_R F_q\ ,\neq 0,1$

$r'[0],\in_R F_q\ ,\neq 0,1$

$v \in_R F^*_p$, $\neq 1$, s.t. $v^q=1$ $/F^*_p$
$\lambda[\mu] \in_R F_q$, $\neq 0,1$ $\mu=0,\ldots,n$
$u \in_R F^*_p$, $\neq 1$, s.t. $u^q=1$ $/F^*_p$
【０１９５】入力暗号文列　$\eta[i,0], \eta[i,1]; i=1,\ldots,$ nと、公開鍵　$\eta[0,\Gamma]; \Gamma=0,1$より、入力文列1000　$g[\mu\Gamma]; \mu=0,\ldots,n; \Gamma=0,1$を、
$g[0,\Gamma]=\eta[0,\Gamma]$　$\Gamma=0,1$
$g[i,\Gamma]=\eta[i,\Gamma]$　$i=1,\ldots,n, \Gamma=0,1$
とする。
【０１９６】以下、証明付再暗号シャッフル方法を用いる。
【０１９７】変換情報保有コミットメント生成処理1042における再暗号シャッフル行列作用1009により、上記暗号シャッフル行列1001を入力文列1000に以下の様に作用させて、出力文列1010　$g''[\mu,\Gamma]; \mu=0,\ldots,n; \Gamma=0,1$を、
$g''[\mu,\Gamma]=\Pi_{\nu=0}{}^{n} g[\nu,\Gamma]^{A[\nu,\mu]}$ $/F^*_p$　　$\mu=0,\ldots,n, \Gamma=0,1$
と生成する。
【０１９８】ここで、$g''[i,\Gamma]; i=1,\ldots,n; \Gamma=0,1$、出力暗号文列1011、$g''[0,\Gamma]; \Gamma=0,1$を第1の変換情報保有コミットメント1012とする。
【０１９９】第２の変換情報保有コミットメント生成処理1044により、さらに入力文列1000から選択1018して、第２の入力文列1019を$g[\mu,\Gamma']$とする。ここでは、$\Gamma'=0$とする。
【０２００】第２の変換情報保有コミットメント1021 $G''[0,\Gamma']$を、
$G''[0,\Gamma']=\Pi_{\nu=0}{}^{n} g[\nu,\Gamma']^{A[\nu,0]}$ $/F^*_p$　$\Gamma=0$or1
と生成1020する。
【０２０１】変換条件コミットメント生成処理1045における恒等式係数計算1022により、元係数1005　ρ', ρ''と、再暗号シャッフル行列1001と、を用いて、恒等式係数1023　$\psi[i], \phi[i], \phi[0], \rho', \rho''; i=1,\ldots,n$を、
$\rho'=\rho'$
$\rho''=\rho''$
$\psi[i]=\Sigma_{j=1}{}^{n}(3A[j,0]+\rho''\lambda[j])A[j,i]$ $/F_q$　　$i=1,\ldots,n$
$\phi[i]=\Sigma_{j=1}{}^{n}(3A[j,0]A[j,0]A[j,i]+2\rho''\lambda[j]A[j,0]A[j,i])+\rho'A[0,i]$ $/F_q$ $i=1,\ldots,n$
$\phi[0]=\Sigma_{j=1}{}^{n}(A[j,0]A[j,0]A[j,0]+\rho''\lambda[j]A[j,0]A[j,0])+\rho''\lambda[0]+\rho'A[0,0]$ $/F_q$
と生成する。
【０２０２】さらに、係数基底1003vを用いて、隠蔽処理1024により、恒等式係数1023 $\phi[0], \rho', \rho''$を、
$\omega=v^{\phi_{[0]}}$ $/F^*_p$
$v''=v^{\rho''}$ $/F^*_p$
$v'=v^{\rho'}$ $/F^*_p$
とコミット1025する。
【０２０３】さらに、係数基底1008uを用いて、準元係数1007 $\lambda[\mu]; \mu=0,\ldots,n$を、
$u[0]=u^{\lambda_{[0]}}$ $/F^*_p$
$u[i]=u^{\lambda_{[i]}}$ $/F^*_p$　　$i=1,\ldots,n$
とコミット1027する。
【０２０４】変換条件コミットメント生成処理1043における恒等式係数計算1013により、元係数1002r'[0]と再暗号シャッフル行列1001と第２の情報隠蔽因子1004とを用いて恒等式係数1014　$\Phi[\nu], r'[0]; \nu=0,\ldots,n$を、
$r'[0]=r'[0]$
$\Phi[0]=\Sigma_{j=1}{}^{n} B[j,0]B[j,0]+r'[0]B[0,0]$ $/F_q$
$\Phi[i]=2\Sigma_{j=1}{}^{n} B[j,0]A[j,i]+r'[0]A[0,i]$ $/F_q$　$i=1,\ldots,n$
と生成する。
【０２０５】さらに係数基底1003　vを用いて、隠蔽処理1015により、恒等式係数1014　$r'[0], \Phi[0]$を、
$V'=v^{r'_{[0]}}$ $/F^*_p$
$\Omega=v^{\Phi_{[0]}}$ $/F^*_p$
とコミット1016する。
【０２０６】以上により、第1の変換条件コミットメント1028を、$\psi[i], \phi[i], \omega, v'', v', v, u, u[0], u[i]; i=1,\ldots,n$とする。第２の変換条件コミットメント1016を、$\Phi[i], V', \Omega, v; i=1,\ldots,n$とする。
【０２０７】ここで、第１のコミットメント1017を、第１の変換情報保有コミットメント1012と第１の変換条件コミットメント1028とし、第２のコミットメント1029を第２の変換情報保有コミットメント1021と第２の変換条件コミットメント1016とする。
【０２０８】応答生成処理1046により、以上の入力文列1000と、出力暗号文列1011と、第1コミットメント1017とを、挑戦値生成関数1030の引数として、第１の挑戦値1031を、
$c[0]=1$
$c[i]=\mathrm{Hash}[i](g[\nu,\Gamma], g''[\nu,\Gamma], u[\nu], u, \phi[j], \psi[j], \omega, v', v'', v; \nu=0,\ldots,n; j=1,\ldots,n; \Gamma=0,1)$　　$i=1,\ldots,n$
と生成し、この挑戦値1031から、再暗号シャッフル行列1001を用いて、第１の応答1033を、
$r[\mu]=\Sigma_{\nu=0}{}^{n} A[\mu,\nu]c[\nu]$ $/F_q$　　$\mu=0,\ldots,n$
と生成1032する。
【０２０９】さらに、準応答1039を、準元係数1007 $\lambda[\mu]; \mu=0,\ldots,n$と、応答1033より、
$r'=\lambda[0]+\Sigma_{i=1}{}^{n}\lambda[i]r[i]r[i]$ $/F_q$
と生成1038する。
【０２１０】応答生成処理1047により、第２の入力文列1019と、出力暗号文列1011と、第２コミットメント1029とを、挑戦値生成関数1034の引数として、第２の挑戦値1035を、
$C[0]=1$
$C[i]=\mathrm{Hash}[i](g[\nu,\Gamma'], G''[0,\Gamma'], g''[j,\Gamma'], \Phi[j], \Omega, V'; \nu=0,\ldots,n; j=1,\ldots,n; \Gamma'=0)$　$i=1,\ldots,n$

と生成し、この挑戦値1035から、再暗号シャッフル行列1001と、第２の情報隠蔽因子1004を用いて、第２の応答1037を、

$R[\mu]=B[\mu,0]+\Sigma_{i=1}{}^{n}A[\mu,i]C[i]\ /F_q \qquad \mu=0,\ldots,n$

と生成1036する。

【０２１１】以上のコミットメント1017、1029と、応答1033、1037と、準応答1039と、を、再暗号シャッフル証明文1040として出力し、再暗号シャッフルの結果として出力暗号文列1011を出力する。

【０２１２】検証方法について、図１１を参照して説明する。

【０２１３】再暗号シャッフル検証方法により、入力文列1000と、出力暗号文列1011と、再暗号シャッフル証明文1040の第1のコミットメント1012、1025、1027を、挑戦値生成関数1100に代入して、第1の挑戦値1101を、

$C[0]=1$

$C[i]=Hash[i]$（入力文列、出力暗号文列、第1のコミットメント）　$i=1,\ldots,n$と生成する。

【０２１４】さらに、第２の入力文列1019と、再暗号シャッフル証明文1040の第２のコミットメント1016、1021と、出力暗号文列1011とを、挑戦値生成関数1108に代入して第２の挑戦値1109を

$c[0]=1$

$c[i]=Hash[i]$（第２の入力文列、出力暗号文列、第２のコミットメント）　$i=1,\ldots,n$

と生成する。

【０２１５】変換情報保有検証処理1112により、第１の挑戦値1101を用いて、入力文列1000と、第１の変換情報保有コミットメント1012と、出力暗号文列1011と、第１の応答1033と、を用いて検証式、

$\Pi_{\mu=0}{}^{n}g[\mu,\Gamma]^{r[\mu]}=\Pi_{\mu=0}{}^{n}g'[\mu,\Gamma]^{c[\mu]}\ /F_p$　$\Gamma=0,1$

が成り立つことを確認1103する。

【０２１６】変換情報保有検証処理1113により、第２の挑戦値1109を用いて第２の入力文列1019と、第２の変換情報保有コミットメント1021と、出力暗号文列1011と、第２の応答1037と、を用いて第２の知識検証式、

$\Pi_{\mu=0}{}^{n}g[\mu,\Gamma']^{R[\mu]}=G'[0,\Gamma']\Pi_{i=1}{}^{n}g'[i,\Gamma']^{c[i]}\ /F_p \qquad \Gamma'=0$

が成り立つことを確認1105する。

【０２１７】変換条件検証処理1111により、第1の挑戦値1101と、第1の応答1033と、第1の変換条件コミットメント1025とを用いて、検証式1102、

$v'^{r'}\ v'^{r[0]}\ \hat{v}\{\Sigma_{i=1}{}^{n}r[i]r[i]r[i]\}=\omega\hat{v}\{\Sigma_{i=1}{}^{n}(c[i]c[i]c[i]+\psi[i]c[i]c[i]+\phi[i]c[i])\}\ /F_p$

と、準応答1039と、準応答コミットメント1027と、第1の応答1033と、検証式1107、

$u^{r'}=u[0]\Pi_{i=1}{}^{n}u[i]^{r[i]r[i]}\ /F_p$

が成り立つことを確認する。

【０２１８】変換条件検証処理1114により、第２の挑戦値1109と、第２の応答1037と、第２の変換条件コミットメント1016とを用いて、検証式1106、

$V'^{r[0]}\ \hat{v}\{\Sigma_{i=1}{}^{n}R[i]R[i]\}=\Omega\hat{v}\{\Sigma_{i=1}{}^{n}(C[i]C[i]+\Phi[i]C[i])\}\ /F_p$

が成り立つことを確認する。

【０２１９】以上全ての検証式が成り立てば証明文を受理1110する。

【０２２０】上記証明付再暗号シャッフル方法は、入力文列に対する再暗号シャッフル行列変換が少なくとも置換行列に属するシャッフル行列を持つ再暗号シャッフル行列により行われたことを保証する効果がある。これは再暗号シャッフルが行われたことを意味し、本実施例は証明付再暗号シャッフルである。

【０２２１】なお証明付再暗号シャッフル装置の変換情報保有コミットメント処理1042、変換条件コミットメント生成処理1043、1045、応答生成処理1046、1047は、コンピュータ上で実行されるプログラムによりその処理・機能が実現される。また証明付再暗号シャッフル検証装置の変換情報保有検証処理1112、1113、変換条件検証処理1111、1114は、コンピュータ上で実行されるプログラムによりその処理・機能が実現される。この場合、該プログラムを記録した記録媒体（例えばＣＤ－ＲＯＭ、ＤＶＤ（digital versatile disk）、フロッピディスク媒体、ハードディスク媒体、磁気テープ媒体、半導体メモリ等のいずれか）から、該プログラムをコンピュータの主記憶にロードして実行することで、本発明を実施することができる。

【０２２２】［実施例(5)］本発明の実施例(5)をなす入力文列生成方法について、図１２を参照して、以下に説明する。Γは、0、1、2の値をとる。

【０２２３】公開鍵302　$g[0,0]$、$g[0,1]$に対応する秘密鍵xは、t人の証明者により分散所持されている。

【０２２４】この時、秘密鍵$x[\Lambda];\Lambda=1,\ldots,t$により、各証明者の公開鍵を、$g[0,0]$、$g[0,1,\Lambda]=g[0,0]^{x[\Lambda]};\Lambda=1,\ldots,t$とし、全体の公開鍵は、$g[0,0]$、$g[0,1]=\Pi_{k=1}{}^{t}g[0,1,\Lambda]$とする。

【０２２５】入力暗号文列301　$\eta[i,0]$、$\eta[i,1];i=1,\ldots,n$と、公開鍵302　$\eta[0,0]$、$\eta[0,1]$が入力され、公開鍵302およびElgamal領域変数p、qより、基底生成関数1200で、入力ベクトル1201を生成し、入力文列300　$g[\mu,\Gamma];\mu=0,\ldots,n;\Gamma=0,1,2$を、

$g[0,\Gamma]=\eta[0,\Gamma] \qquad \Gamma=0,1$

$g[i,\Gamma]=\eta[i,\Gamma]/F_p \quad i=1,\ldots,n, \quad \Gamma=0,1$

$g[\mu,2]=Hash'[\mu](p,q,\eta[0,0],g[0,1,\Lambda];\Lambda=1,\ldots,t)\ \mu=0,\ldots,n$

とする。

【０２２６】前記実施例(1)から実施例(4)に、本実施例の入力文列生成方法を適用する場合、これらの実施例におけるΓの値をとる範囲を、全て0、1から、0、1、2に

変更する。この新たに導入された、入力暗号文でも公開鍵でもないΓ＝２の成分が、入力暗号文生成者にも意図できない入力文列の成分となり、証明者が生成できる応答に制限を課す働きをし、入力暗号文生成者と再暗号シャッフル証明文生成者とが共謀して再暗号シャッフル証明文の偽造を行うことを阻止する。

【０２２７】また前記実施例(3)に、本実施例の入力文列生成方法を適用する場合、入力文列を、g[-1, Γ]まで拡張して、公開鍵　g[-1,0]、g[-11]、η[0,0]、η[0,1]より、

g[-1, Γ]=η[-1, Γ]　　Γ=0,1

g[0, Γ]=η[0, Γ]　　Γ=0,1

g[i, Γ]=η[i, Γ] /F*p,　　i=1,...,n, Γ=0,1

g[μ,2]=Hash'[μ](p,q, η[0,0],g[0,1,Λ];Λ=1,...,t)　μ=-1,...,n

とする。

【０２２８】また実施例(4)に、本実施例の入力文列生成方法を適用する場合、Γ'＝２とし、第２の情報隠蔽因子より、第２の変換情報保有コミットメントを、

C''[02]=Πν=1n g[ν,2]A'[ν,0] /F*p

C''[i2]=Πν=1n g[ν,2]A[ν,i] /F*p,　　i=1,...,n

と変更する。

【０２２９】さらに、証明付再暗号シャッフル方法または再暗号シャッフル検証方法において、第２の入力文列　g[μ,Γ'];　Γ＝２と、第２のコミットメントとを挑戦値生成関数の引数として、第２の挑戦値を、

C[0]=1

C[i]=Hash[i](g[ν,2],C''[ν,2],Φ[j],Ω,V';;ν=0,...,n;j=1,...,n;Γ=0,1)　i=1,...,n

と変更する。

【０２３０】さらに、変換情報保有検証処理における第２の知識検証式は、

Πμ=0n g[μ,2]r[μ]=Πμ=0n C''[μ,Γ']c[μ] /F*p

と変更する。

【０２３１】［実施例(6)］本発明の実施例(6)をなす入力文列生成方法について、図１３と図１４を参照して説明する。Γは、0、1の値をとる。

【０２３２】前記実施例(5)と同様に、秘密鍵xは、t人の証明者により分散所持されている。各証明者Λ;Λ=1,...,tは、証明付公開鍵列方法1304により、入力暗号文列301η[i,0]、η[i,1];i=1,...,nと、公開鍵302　η[0,0]、η[0,1]とを、共通初期値1310として、秘密鍵1301x[Λ]と、疑似秘密鍵1302α[Λ]と、を公開鍵列情報1300として入力し、分散公開鍵列対1305　g'[μ,1,Λ];μ=0,...,nと、公開鍵列証明文1306と、を得る。

【０２３３】公開鍵列検証方法1307により、各証明者の出力した分散公開鍵列対1305と、公開鍵列証明文と、共通初期値1310とから、分散公開鍵列1305の正当性が検証されたら、前処理方法により、各証明者の分散公開鍵列対1305　g'[μ,1,Λ];μ=0,...,n;Λ=1,...,tを合わせて、公開鍵列対140　3g'[μ,1,Λ];μ=0,...,nを、

g'[μ,1]=ΠΛ=1t g'[μ,1,Λ] /F*p,　　μ=0,...,n

とする。ここで、g'[0,1]=η[0,1]に入れ替える。

【０２３４】共通初期値である入力暗号文列301　η[i,0]、η[i,1];i=1,...,nと、公開鍵302η[0,0]、η[0,1]とから、公開鍵列底1401　g'[μ0];μ=0,...,nを、

g'[0,0]=η[0,0]

g'[i,0]=Hash'[i](η[0,0],η[0,1,Λ],η[j,Γ];Λ=1,...,t;Γ=0,1;j=1,...,n;)　i=1,...,n

と生成1400する。ここでも、公開鍵列対1403と同様にg'[0,0]が入れ替えている。

【０２３５】公開鍵列底1401と公開鍵列対1403を合わせて、公開鍵列1404　g'[μ,Γ];μ=0,...,n,Γ=0,1とする。

【０２３６】公開鍵列1404と、入力暗号文列301と、公開鍵302とから、入力文列300　g[μ,Γ];μ=0,...,n;Γ=0,1を、

g[0,Γ]=η[0,Γ]　　Γ=0,1

g[i,Γ]=η[i,Γ]g'[i,Γ] /F*p,　　i=1,...,n,Γ=0,1

とする（前処理1402）。

【０２３７】前記実施例(3)に、本実施例の入力文列生成方法を適用する場合、入力暗号文列　η[i,Γ];i=1,...,n;Γ=0,1と、公開鍵　η[0,Γ];Γ=0,1に対して、公開鍵列　g'[μ,Γ];μ=-1,...,n;Γ=0,1を生成する。ただし、g'[0,Γ];Γ=0,1は、公開鍵に等しい。そして、入力文列　g[μ,Γ];μ=-1,...,n;Γ=0,1を、

g[-1,Γ]=η[0,Γ]　　Γ=0,1

g[i,Γ]=η[i,Γ]g'[i,Γ] /F*p,　　i=0,...,n,Γ=0,1

とする。

【０２３８】本実施例では、新たに生成された公開鍵列が、入力暗号文生成者にも意図できないため、入力暗号文にそれを乗じた入力文列の成分も意図できない。そのため、証明者が生成できる応答に制限を課す働きをなし、入力暗号文生成者と再暗号シャッフル証明文生成者とが共謀して再暗号シャッフル証明文の偽造を行うことを阻止する。

【０２３９】なお図１３に示した、証明付公開鍵列装置1304、前処理装置1309、公開鍵列検証装置1307は、コンピュータ上で実行されるプログラムによりその処理・機能が実現される。この場合、該プログラムを記録した記録媒体（例えばＣＤ－ＲＯＭ、ＤＶＤ（digital versatile disk）、フロッピディスク媒体、ハードディスク媒体、磁気テープ媒体、半導体メモリ等のいずれか）から、該プログラムをコンピュータの主記憶にロードして実行することで、本発明を実施することができる。

【０２４０】［実施例(7)］本発明の実施例(7)の入力文列生成方法について、図１５乃至図１８を参照て説明する。Γは、0、1の値をとる。前記実施例(5)と同様に、秘密鍵1502　xは、t人の証明者により、分散所持されている。

【０２４１】各証明者Λ；Λ=1,...,tは、証明付公開鍵列方法1504により、Elgamal領域変数を共通初期値1500として、秘密鍵1502　x[Λ]と、疑似秘密鍵1503　α[Λ]と、を、公開鍵列情報1501として入力し、分散公開鍵列対1505　g'[μ,1,Λ]；μ=0,...,nと、公開鍵列証明文1506と、を得る。

【０２４２】公開鍵列検証方法1507により、各証明者の出力した分散公開鍵列対1505と、公開鍵列証明文1506と、共通初期値1500より、分散公開鍵列対1505の正当性が検証1508されたら、各証明者の分散公開鍵列対150　5 g'[μ,1,Λ]；μ=0,...,n；Λ=1,...,tを合わせて、公開鍵列対1509　g'[μ,1,Λ]；μ=0,...,nを、

$g'[\mu,1]=\Pi_{\Lambda=1}{}^{t} g'[\mu,1,\Lambda]\ /F^{*}{}_{p}$　　μ=0,...,n

とする。

【０２４３】共通初期値1500から公開鍵列底g'[μ,0]；μ=0,...,nを、

$g'[\mu,0]=Hash'[\mu](p,q)$　　μ=0,...,n

と生成する。

【０２４４】公開鍵列底と公開鍵列対1509を合わせて、公開鍵列1611　g'[μΓ]；μ=0,...,n,Γ=0,1とする。

【０２４５】各入力暗号文生成者i=1....,nは、証明付暗号化方法1606により、平文1602m[i]と、個別公開鍵1601g'[i,Γ]；Γ=0,1と、秘密乱数1604s[i]と、疑似秘密乱数1605s'[i]とより、入力暗号文1607η[i,Γ]；Γ=0,1を、

$\eta[i,0]=g'[i,0]^{s[i]}\ /F^{*}{}_{p}$

$\eta[i,1]=m[i]g'[i,1]^{s[i]}\ /F^{*}{}_{p}$

と生成する。

【０２４６】またコミットメント(疑似暗号文底1704)、挑戦値1707、応答1709を順に以下のように生成し、

$\eta[i2]=g'[i,0]^{s'[i]}\ /F^{*}{}_{p}$

$c'[i]=Hash[0](\eta[i,0],\eta[i,1],\eta[i2])$

$\theta'[i]=c'[i]s[i]+s'[i]\ /F_{q}$

疑似暗号文底1704と応答1709とを暗号化証明文1608とする。

【０２４７】暗号化検証装置により、全ての入力暗号文1607と暗号化証明文1608に関して、

$c'[i]=Hash[0](\eta[i,0],\eta[i,1],\eta[i2])$

と、挑戦値1801を求め、これに応答1709を用いて、検証式1802

$\eta[i,0]^{\theta'[i]}=\eta[i,1]^{c'[i]}\eta[i2]\ /F^{*}{}_{p}$

が成り立つことを確認1610する。入力暗号文1607全ての正当性が確認されたら、入力暗号文323　η[i,Γ]；Γ=0,1と、共有公開鍵1600　g'[0,Γ]；Γ=0,1とより入力文列300を、

g[0,Γ]=g'[0,Γ]

g[i,Γ]=η[i,Γ]　i=1,...,n

とする。

【０２４８】前記実施例(3)に、本実施例の入力文列生成方法を適用する場合は、

g[-1Γ]=g'[-1,Γ]

g[0,Γ]=g'[0,Γ]

g[i,Γ]=η[i,Γ]　i=1,...,n

とする。

【０２４９】本実施例では、始めに生成された公開鍵列が、入力暗号文生成者にも意図できないため、これを基に暗号化したことが主命されている入力暗号文の成分も、意図できない。そのため、証明者が生成できる応答に制限を課す働きをし、入力暗号文生成者と再暗号シャッフル証明文生成者とが共謀して再暗号シャッフル証明文の偽造を行うことを阻止する。

【０２５０】なお図１５に示した、証明付公開鍵列装置1504、公開鍵列検証装置1507は、コンピュータ上で実行されるプログラムによりその処理・機能が実現される。また図１６乃至図１８に示した、証明付暗号化装置1606、暗号化検証装置1609は、コンピュータ上で実行されるプログラムによりその処理・機能が実現される。この場合、該プログラムを記録した記録媒体（例えばＣＤ－ＲＯＭ、ＤＶＤ（digital versatile disk）、フロッピディスク媒体、ハードディスク媒体、磁気テープ媒体、半導体メモリ等のいずれか）から、該プログラムをコンピュータの主記憶にロードして実行することで、本発明を実施することができる。

【０２５１】［実施例(8)］本発明の実施例(8)における証明付公開鍵列方法について図１９と図２０とを参照して説明する。

【０２５２】共通初期値eと、秘密鍵1902　xと、疑似秘密鍵1903　αとが、公開鍵列情報1901として、入力される。

【０２５３】共通初期値1900から、公開鍵列底1905　g'[μ0]；μ=0,...,nを、

$g'[\mu,0]=Hash'[\mu](e)\quad \mu=0,...,n$

と生成1904する。

【０２５４】これから、秘密鍵1902　xと、疑似秘密鍵1903αとにより、(分散)公開鍵列対1907　g'[μ,1]；μ=0,...,nが、

$g'[\mu,1]=g'[\mu,0]^{x}\ /F^{*}{}_{p}$　　μ=0,...,n

と生成1906され、疑似公開鍵列対1909が、

$g'[\mu,2]=g'[\mu,0]^{\alpha}\ /F^{*}{}_{p}$　　μ=0,...,n

と生成1908される。

【０２５５】挑戦値1912、応答1914を順に、

$c''=Hash[0](g'[\mu,0],g'[\mu,2]\ ;\ \mu=0,...,n)$

$\theta=c''x+\alpha\ /F_{q}$

と生成し、疑似公開鍵列1909と応答1914を公開鍵列証明文1915とする。

【０２５６】公開鍵列検証方法により、挑戦値2003を、

$c''=Hash[0](g'[\mu,0],g'[\mu,2];\mu=0,...,n)$

と生成2000し、応答1914を用いて検証式、

$g'[\mu,0]^{\theta}=g'[\mu,0]^{c''}g'[\mu,2]\ /F^{*}{}_{p}$　　μ=0,...,n

を検証2004する。

【０２５７】本実施例では、始めに生成された公開鍵列底、誰にも意図できないため、これを元に作られた公開鍵列の成分も意図できない。

【０２５８】なお図１９、図２０に示した、証明付公開鍵列装置、公開鍵列検証装置は、コンピュータ上で実行されるプログラムによりその処理・機能が実現される。この場合、該プログラムを記録した記録媒体（例えばＣＤ－ＲＯＭ、ＤＶＤ（digital versatile disk）、フロッピディスク媒体、ハードディスク媒体、磁気テープ媒体、半導体メモリ等のいずれか）から、該プログラムをコンピュータの主記憶にロードして実行することで、本発明を実施することができる。

【０２５９】［実施例(9)］本発明の実施例(9)として、証明付復号について説明する。前記実施例(5)と同様に、秘密鍵　xは、t人の証明者により分散所持されている。

【０２６０】Λ；Λ=1,...,t番目の証明者は、Λ-1番目の証明者による部分復号の結果を入力して、それを部分復号する。Λ番目の証明者による部分復号の結果が復号文である。ただし、0番目の証明者による部分復号の結果は、上記最終的な再暗号シャッフルの出力のことである。

【０２６１】ここで、Λ番目の証明者の行う証明付部分復号(部分復号およびその証明文の提出)について説明する。

【０２６２】疑似乱数発生器により1,0でないF_q上の数を$\beta[\Lambda]$を作成する。

【０２６３】$\beta[\Lambda]\in_R F_q, \neq 0,1$

【０２６４】また、自身の公開鍵　g[0,0]、g'[0,1,Λ]を、g[0,0]、g[0,1]と、入力された暗号文列を、g[i,Γ]；i=1,...,n,　Γ=0,1とし、自身の公開鍵と秘密鍵x[Λ]から、部分復号基底G[μ,0Λ]；μ=0,...,nと、疑似部分復号基底G[μ,1,Λ]；μ=0,...,nを、

$G[\mu,0,\Lambda]=g[\mu,0]^{x[\Lambda]}\ /F_p^*\quad \mu=0,\ldots,n$　　＊

$\Pi_{\mu=1}^{n+n'} g[\mu,\Gamma]^{r[\mu]}=\Pi_{\mu=1}^{n+n'} g[\mu,\Gamma]^{\wedge\{\Sigma_{\nu=1}^{n+n'} A[\mu,\nu]c[\nu]\}}\ /F_p^*$

$=\Pi_{\nu=1}^{n+n'}(\Pi_{\mu=1}^{n+n'} g[\mu,\Gamma]^{A[\mu,\nu]})^{c[\nu]}\ /F_p^*$

$=\Pi_{\nu=1}^{n+n'} g''[\nu,\Gamma]^{c[\nu]}\ /F_p^*$

より分かる。　　※よりわかる。

【０２７２】準元係数をコミットしたものと、これに付随する応答と準応答が検証式を満たすことは、

$u'^{r'}=u\hat{}\{\lambda[0]+\Sigma_{i=1}^{n}\lambda[i]r[i]r[i]\}\ /F_p^*$

$=u^{\lambda[0]}\Pi_{i=1}^{n}(u^{\lambda[i]})^{r[i]r[i]}\ /F_p^*$

$=u[0]\Pi_{i=1}^{n}u[i]^{r[i]r[i]}\ /F_p^*$　　※

$v'^{r[0]}\Pi_{i=1}^{n}v^{r[i]r[i]}\ /F_p^*$

$=(v'^{r'[0]})^{r[0]}\ v\hat{}\{\Sigma_{i=1}^{n}\Sigma_{\mu=0}^{n}\Sigma_{\nu=0}^{n}A[i,\mu]A[i,\nu]c[\mu]c[\nu]\}\ /F_p^*$

$=v\hat{}\{r'[0]\Sigma_{\mu=0}^{n}[0,\mu]c[\mu]+2\Sigma_{i=1}^{n}\Sigma_{j=1}^{n}A[i,0]A[i,j]c[j]+\Sigma_{i=1}^{n}A[i,0]A[i,0]+\Sigma_{i=1}^{n}\Sigma_{j=1}^{n}\Sigma_{k=1}^{n}A[i,j]A[i,k]c[j]c[k]\}\ /F_p^*$

$=v\hat{}\{\Sigma_{i=1}^{n}\phi[i]c[i]+\phi[0]+\Sigma_{i=1}^{n}c[i]c[i]\}\ /F_p^*$

$=\omega v\hat{}\{\Sigma_{i=1}^{n}(c[i]c[i]+\phi[i]c[i])\}\ /F_p^*$

＊$G[\mu,1,\Lambda]=g[\mu,0]^{\beta[\Lambda]}\ /F_p^*\quad \mu=0,\ldots,n$

と生成する。コミットメントとして、g[μ,ΓΛ]；μ=0,...,n,Γ=0,1,Λ=0,...,tを出力する。

【０２６５】$g[0,1,\Lambda]=g[0,0]^{x[\Lambda]}=G[0,0,\Lambda]$は、公開鍵と重複しているが同じものが計算されている。

【０２６６】挑戦値を、

$c[\Lambda]=Hash[0](g[\mu,0],G[\mu,\Gamma,\Lambda];\mu=0,\ldots,n;\Gamma=0,1)$

と生成し、これを用いて、応答r[Λ]を

$r[\Lambda]=\beta[\Lambda]+c[\Lambda]x[\Lambda]\ /F_q$

と生成して出力する。部分復号基底、疑似部分復号基底と応答を証明付部分復号の証明文として出力する。

【０２６７】部分復号を

$g[i,0]\rightarrow g[i,0]\quad i=1,\ldots,n$

$g[i,1]\rightarrow g[i,1]/G[i,0,\Lambda]\ /F_p^*,\quad i=1,\ldots,n$

として出力する。

【０２６８】検証処理は、入力暗号文列と証明文中とより挑戦値を、

$c[\Lambda]=Hash[0](g[\mu,0],G[\mu,\Gamma,\Lambda];\mu=0,\ldots,n;\Gamma=0,1)$

と生成し、証明文中の応答、入力暗号文列、分復号基底、疑似部分復号基底を用いて、

$g[\mu,0]^{r[\Lambda]}=G[\mu,0,\Lambda]^{c[\Lambda]}G[\mu,1,\Lambda]\ /F_p^*\quad \mu=0,\ldots,n$

を確認し、さらに部分復号が、このG[μ,0,Λ]を用いて行われたことを確認して受理する。

【０２６９】以上を、t人の証明者全てにより行われた結果を復号文とする。

【０２７０】［正当性］以上説明した実施例の正当性について説明する。

【０２７１】［完全性］入力文列と、出力暗号文列と変換情報保有コミットメントとである出力文列と、これに付随する応答と挑戦値が変換情報保有検証処理の検証式を満たすことは、

$g[\mu,\Gamma]\hat{}\{\Sigma_{\nu=1}^{n+n'}A[\mu,\nu]c[\nu]\}$

※よりわかる。

【０２７３】変換条件コミットメント生成処理が出力した恒等式の係数と、これに付随する応答と挑戦値が知識検証処理の検証式を満たすことは、以下のようにして分かる。

【０２７４】実施例(1)の恒等式係数に関しては、

より成立することがわかる。以上、A[i,j]が置換行列であるという事実を使った。

【０２７５】前記実施例(2)の恒等式係数に関しては、

$$\Sigma_{i=1}^{n} r[i]r[i]r[i]+\Sigma_{i=1}^{n}\rho''\lambda[i]r[i]r[i]+\rho' r[0]$$
$$=\Sigma_{h=1}^{n}\Sigma_{i=1}^{n}\Sigma_{j=1}^{n}\Sigma_{k=1}^{n} A[h,i]A[h,j]A[h,k]c[i]c[j]c[k]$$
$$+\Sigma_{h=1}^{n}\Sigma_{i=1}^{n}\Sigma_{j=1}^{n}(3A[h,0]A[h,i]A[h,j]+\rho''\lambda[h]A[h,i]A[h,j])c[i]c[j]$$
$$+\Sigma_{h=1}^{n}\Sigma_{i=1}^{n}(3A[h,0]A[h,0]A[h,i]+2\rho''\lambda[h]A[h,0]A[h,i]+\rho' A[0,i])c[i]$$
$$+\Sigma_{h=1}^{n}(A[h,0]A[h,0]A[h,0]+\rho''\lambda[h]A[h,0]A[h,0])+\rho''\lambda[0]+\rho' A[0,0]$$

$$/F_q$$
$$=\Sigma_{h=1}^{n}(c[h]c[h]c[h]+\psi[h]c[h]c[h]+\phi[i]c[i]+\phi[0])\quad /F_q$$

であり、これは、$v^{\wedge}\{\Sigma_{h=1}^{n}(c[h]c[h]c[h]+\psi[h]c[h]c[h]+\phi[i]c[i])\}\omega[0]\quad /F_p$の指数部と等しい。

【０２７６】以上で最後の式を導くためにA[i,j]が置換行列であるという事実を使った。

【０２７７】前記実施例(3)、及び実施例(4)に関しても同様の議論より分かる。

【０２７８】前記実施例(8)の証明付公開鍵列方法が出力した公開鍵列底と、公開鍵列対と疑似公開鍵列対と、これに付随する応答と挑戦値が検証処理の検証式を満たすことは、

$g'[\mu,0]^{r}=g'[\mu,0]^{\alpha x+\alpha}\quad /F_p^*$

$=g'[\mu,0]^{\alpha x}g'[\mu,0]^{\alpha}\quad /F_p^*$

$=g'[\mu,1]^{c}g'[\mu,2]\quad /F_p^*$

から分かる。

【０２７９】［健全性］与えられた挑戦値$c[\nu];\nu=1,\ldots,n+m'$に対する、変換情報保有検証処理における検証式を満たす応答$r[\mu];\mu=1,\ldots,n+m$を求めるには$A[\mu\ \nu];\mu=1,\ldots,n+m;\nu=1,\ldots,n+m'$を知らなければならない。

【０２８０】これは、与えられた$g[\mu,\Gamma]$、$g''[\nu\ \Gamma];\mu=1,\ldots,n+m;\nu=1,\ldots,n+m'$に対して、$A[\mu\ \nu];\mu=1,\ldots,n+m;\nu=1,\ldots,n+m'$を知らずして、等価検証処理における検証式を満たす応答を求めることは、離散対数問題を解くことに等しいからである。

【０２８１】なぜならば、$A[\mu,\nu]$を知らないと言う事は、少なくとも一つの$g''[\nu,\Gamma]$に関しては、$g[\mu,\Gamma];\mu=1,\ldots,n+m$を基底とするその表現を知らない。その時、任意のcに関して、検証式を満たす応答を求めることができるなら、$c[\xi]=1$、$c[\nu]=0;\nu=0,\ldots,\xi-1,\xi+1,\ldots,n+m'$なる$c[\nu]$を選ぶ事で、離散対数を解くことが出来るからである。

【０２８２】また、挑戦値$c[\nu]$は、コミットメント$g[\mu,\Gamma],g''[\mu,\Gamma]$を引数に持つため、挑戦値が決定してから、コミットメントを調節することができない(挑戦値生成関数がこの性質を持つことを要求する)。そのため、証明者には、挑戦値はコミットメント決定後に与えられた乱数と考えることができる。

【０２８３】どの$g[\mu,\Gamma]$の成分に関しても、他の成分を基底とするその表現を知らなければ、検証式を満たす応答を複数作る事は、離散対数問題を解く事に等しい。なぜなら、異なる$r[\mu]$、$r'[\mu]$に関して、検証式が成り立つならば、両辺を互いに割る事により、$g[\mu,\Gamma]$を基底とする、非自明な1の表現が得られる。これは離散対数問題を解く事に等しいからである。

【０２８４】入力文列生成方法により生成される入力文列　$g[\mu\ \Gamma];\mu=1,\ldots,n+m;\Gamma=0,\ldots$は、いずれかの$\Gamma$に関してなすベクトル　$g[\mu,\Gamma];\mu=1,\ldots,n+m$が、ハッシュ関数で生成されているか、あるいはハッシュ関数で生成されているベクトルを乗じた等の操作で生成されていることが明らかになっているため、互いに、他を基底として表現する事が計算量的に困難である、と考えられる。

【０２８５】以上により、検証式を満たす$r[\mu];\mu=1,\ldots,n+m$として、$g''[\nu,\Gamma]=\Pi_{\mu=1}^{n+m} g[\mu,\Gamma]^{A[\mu\ \nu]}\quad /F_p^*;\nu=1,\ldots,n+m'$なる$A[\mu,\nu]$を用いて、$r[\mu]=\Sigma_{\nu=1}^{n+m'}A[\mu,\nu]c[\nu]\quad /F_q;\mu=1,\ldots,n+m$、と生成する以外には、証明者は計算できない。個別公開鍵を用いる方法でも同様である。

【０２８６】上述のようにあるΓに関して

$g''[\nu,\Gamma]=\Pi_{\mu=1}^{n+m} g[\mu,\Gamma]^{A[\mu\ \nu]}\quad /F_p^*\qquad \nu=1,\ldots,n+m'$

の関係が証明されたならば、他のΓに関しても以下のように同様に証明される。

【０２８７】挑戦値生成関数の引数に含めた$g[\mu,\Gamma],g''[\nu,\Gamma]$に対して検証式が成り立つならば、

$g''[\nu,\Gamma]=\Pi_{\mu=1}^{n+m} g[\mu,\Gamma]^{A[\mu\ \nu]}\quad /F_p^*\qquad \nu=1,\ldots,n+m'$

である。

【０２８８】なぜならば、

$g''[\nu,\Gamma]=\Pi_{\mu=1}^{n+m} g[\mu,\Gamma]^{A'[\mu\nu]}\quad /F_p^*\qquad \nu=1,\ldots,n+m'$

と表した時に検証式が成り立つならば、

$=\Pi_{\mu=1}^{n+m} g[\mu,\Gamma]^{\wedge}\{\Sigma_{\nu=1}^{n+m'}(A[\mu,\nu]-A'[\mu,\nu])c[\nu]\}=1\quad /F_p^*$

が成り立つ。

【０２８９】ところが、無作為に選ばれた$c[\nu]$に関して、これが成り立つのは、

$\Pi_{\mu=1}^{n+m} g[\mu,\Gamma]^{A[\mu\ \nu]}=\Pi_{\mu=1}^{n+m} g[\mu,\Gamma]^{A'[\mu\ \nu]}\quad /F_p^*\qquad \nu=1,\ldots,n+m'$

の時だけであるからである。

【０２９０】前記実施例(2)において、変換条件コミットメント生成処理により準元係数をコミットしたものu、u[μ];μ=0,...,nが与えられたとき、応答r[i];i=1,...,nと、準応答r'が検証式を満たす時、準応答r'は一意であり、

$r'=\lambda[0]+\Sigma_{i=1}^{n}\lambda[i]r[i]r[i]\ /F_q$

が、検証式を満たすことよりr'は上式で表されるものである。

【０２９１】前記実施例(2)の恒等式の検証式の左辺のvの指数部を展開すると、

$\Sigma_{h=1}^{n}\Sigma_{i=1}^{n}\Sigma_{j=1}^{n}\Sigma_{k=1}^{n}A[h,i]A[h,j]A[h,k]c[i]c[j]c[k]+\Sigma_{h=1}^{n}\Sigma_{i=1}^{n}\Sigma_{j=1}^{n}(3A[h,0]A[h,i]A[h,j]+\rho''\lambda[h]A[h,i]A[h,j])c[i]c[j]+\Sigma_{i=1}^{n}(\Sigma_{h=1}^{n}(3A[h,0]A[h,0]A[h,i]+2\rho''\lambda[h]A[h,0]A[h,i])+\rho'A[0,i])c[i]+\Sigma_{h=1}^{n}(A[h,0]A[h,0]A[h,0]+\rho''\lambda[h]A[h,0]A[h,0])+\rho''\lambda[0]+\rho'A[0,0]\ /F_q$

である。

【０２９２】右辺のvの指数部は、

$\Sigma_{i=1}^{n}(c[i]c[i]c[i]+\psi[i]c[i]c[i]+\phi[i]c[i])+\phi[0]\ /F_q$

である。

【０２９３】よって、いかなるc[μ];μ=0,...,nに関しても、検証式が成り立つためには、c[μ]c[ν]c[ξ];μ,ν,ξ=0,...,nの係数が同じでなければならない。それ以外の時に、検証式が無作為に与えられたc[μ]に関して成り立つ可能性は、無視できる。

【０２９４】これは、

$\delta[i,j]=1$　　i=jの場合

$=0$　　その他

$\delta'[i,j,k]=1$　i=j=kの場合

$=0$　　その他

を用いて、

$\Sigma_{h=1}^{n}A[h,i]A[h,j]A[h,k]=\delta'[i,j,k]\ /F_q$　i,j,k=1,...,n

$\Sigma_{h=1}^{n}(3A[h,0]A[h,i]A[h,j]+\rho''\lambda[h]A[h,i]A[h,j])=\delta[i,j]\psi[i]\ /F_q$　i,j=1,...,n

$\Sigma_{h=1}^{n}(3A[h,0]A[h,0]A[h,i]+2\rho''\lambda[h]A[h,0]A[h,i])+\rho'A[0,i]=\phi[i]\ /F_q$ i=1,...,n

$\Sigma_{h=1}^{n}(A[h,0]A[h,0]A[h,0]+\rho''\lambda[h]A[h,0]A[h,0])+\rho''\lambda[0]+\rho'A[0,0]=\phi[0]\ /F_q$

であることを保証する。このことから、A[i,j];i,j=1,...,nについて以下のことが分かる。

【０２９５】与えられたj,k;j≠kに対して、第h成分がA[h,j]A[h,k]であるn次元ベクトルA[h,j]A[h,k];h=1,...,nと、与えられたiに対して第h成分がA[h,i]であるn次元ベクトルA[h,i];h=1,...,nとを考える。今、n個のベクトルA[h,i];i=1,...,nがn次元空間を張る、すなわち全てのベクトルがA[h,i];i=1,...,nの線形結合で表せるとする。すると、上式よりベクトルA[h,j]A[h,k];h=1,...,nは、すべてのベクトルA[h,i]と内積が0であるので、

$A[h,j]A[h,k]=0\ /F_q$　　h=1,...,n

である。

【０２９６】このことから、n個のベクトルA[h,i];h=1,...,n;i=1,...,nのうち、各h成分が0でないベクトルはひとつしかない。

【０２９７】また上式より、i=j=kの時、A[h,i]A[h,j]A[h,k]≠0であるため、ベクトルA[h,i];h=1,...,nは少なくともひとつは0でない成分を持つ。よって、全てのベクトルA[h,i];h=1,...,nは、0でない成分をただひとつ持ち、上式より、それは$1^{1/3}$である。

【０２９８】今度は、n個のベクトルA[h,i];h=1,...,n;i=1,...,nがn次元空間を張ることを示す。

【０２９９】ベクトルa[h];h=1,...,nを、n個のスカラーκ[i];i=1,...,nを用いて、

$a[h]=\Sigma_{i=1}^{n}\kappa[i]A[h,i]$　　h=1,...,n　$/F_q$

と表す。

【０３００】もし、a[h]=$/F_q$ならば、κ[i]=0であることを示せば、n個のベクトルA[h,i];h=1,...,n;i=1,...,nがn次元空間を張ることを示せる。a[h]=0　$/F_q$ならば、上式の両辺に、第h成分がA[h,i]A[h,i]であるn次元ベクトルA[h,i]A[h,i]を掛けると、上式の二つ上の式より、

$0=\kappa[i]\ /F_q$　　i=1,...,n

となる。以上をもってA[i,j]が置換行列であるか、置換行列のいくつかの成分に$1^{1/3}$を乗じることによって得られる準置換行列であることが示された。

【０３０１】実施例(1)の恒等式の検証式の左辺のvの指数部を展開すると、

$r[0]r[0]+\Sigma_{i=1}^{n}r[i]r[i]\ /F_q$

$=\Sigma_{i=1}^{n}\Sigma_{j=1}^{n}\Sigma_{k=1}^{n}A[i,j]A[i,k]c[j]c[k]$

$+\Sigma_{j=1}^{n}(\Sigma_{i=1}^{n}2A[i,0]A[i,j]+r'[0]A[0,j])c[j]$

$+\Sigma_{i=1}^{n}A[i,0]A[i,0]+r'[0]A[0,0]\ /F_q$

である。右辺のvの指数部は、

$\Sigma_{i=1}^{n}(c[i]c[i]+\phi[i]c[i])+\phi[0]\ /F_q$

である。

【０３０２】よっていかなるc[μ];μ=0,...,nに関しても、検証式が成り立つためには、c[μ]cν;μ,ν=0,...,nの係数が同じでなければならない。それ以外の時に検証式が無作為に与えられた応答に関して成り立つ可能性は無視できる。

【０３０３】これは

$\Sigma_{h=1}^{n}A[h,i]A[h,j]=\delta[i,j]\ /F_q$

$\phi[i]=\Sigma_{h=1}^{n}2A[h,0]A[h,i]+r'[0]A[0,i]\ /F_q$

$\phi[0]=\Sigma_{i=1}^{n}A[i,0]A[i,0]+r'[0]A[0,0]\ /F_q$

であることを保証する。よって、A[i,j];i,j=1,...,nは正規直交行列でなければ検証式は成り立つ可能性は無視できる。

【０３０４】前記実施例(3)、実施例(4)においても同様の議論が成り立ち、A[i,j];i,j=1,...,nは、置換行列か

つ正規直交行列である。そして、これは置換行列であることを意味する。

【０３０５】［知識隠匿性］再暗号シャッフル証明文において、再暗号シャッフルの情報が計算量的に秘匿されていることを示す。

【０３０６】再暗号シャッフルの結果g''[ν],m''[μ]以外にも、

r[μ],r',φ[i],ψ[i],ω,v',v'',v,u,u[i],R[μ],Φ[i],V',Ω,v

等の値が明らかになる。これらは再暗号シャッフルに関する情報を与えている。しかし指数演算された結果でない条件の数より、再暗号シャッフル行列演算に関する未知数のほうが多くなるように、恒等式の係数をコミットして隠蔽すれば離散対数問題をといて条件の数を増やさない限り解けない。ただし変数の数だけでなく未知数の条件の中への現れ方で解ける場合もあるので、若干の調整を行う必要がある。

【０３０７】

【発明の効果】以上説明したように、本発明によれば、証明付再暗号シャッフルの計算量を、従来の技術と比べて、低減する、という効果を奏する。

【０３０８】特に、検証処理は事前に計算しておくことができない応用例が多いと考えられるので、検証の計算量を比較すると、入力暗号文の数をnとしたとき、従来の技術(1)では、安全変数が160の時冪乗剰余演算が320n+2n回必要とされており、従来の技術(2)では冪乗剰余演算が8(n log n −n+1)回必要とされていたのに対して、本発明によれば、冪乗剰余演算は7n+14回で済み、n>4の時は、いずれの従来の技術よりも冪乗剰余演算が少ない。

【０３０９】しかも、本発明においては、検証過程で行う冪乗剰余は、個別の冪乗剰余演算ではなく、冪乗剰余演算の積の計算であることから、個別の冪乗剰余演算よりも少ない計算量で計算できるため、更なる高速化が望める、という効果を奏する。

【図面の簡単な説明】

【図１】従来の技術１の構成を示す図である。

【図２】従来の技術２の構成を示す図である。

【図３】本発明の実施例における証明付再暗号シャッフル装置と再暗号シャッフル検証装置との情報の入出力を示す図である。

【図４】本発明の実施例１の証明付再暗号シャッフル装置の詳細を示す図である。

【図５】本発明の実施例１の再暗号シャッフル検証装置の詳細を示す図である。

【図６】本発明の実施例２の証明付再暗号シャッフル装置の詳細を示す図である。

【図７】本発明の実施例２の再暗号シャッフル検証装置の詳細を示す図である。

【図８】本発明の実施例３の証明付再暗号シャッフル装置の詳細を示す図である。

【図９】本発明の実施例３の再暗号シャッフル検証装置の詳細を示す図である。

【図１０】本発明の実施例４の証明付再暗号シャッフル装置の詳細を示す図である。

【図１１】本発明の実施例４の再暗号シャッフル検証装置の詳細を示す図である。

【図１２】本発明の実施例５の入力文列生成装置の詳細を示す図である。

【図１３】本発明の実施例６の入力文列生成方法の情報の入出力を示す図である。

【図１４】本発明の実施例６における前処理装置の詳細を示す図である。

【図１５】本発明の実施例７の入力文列生成方法の情報の入出力を示す図である。

【図１６】本発明の実施例７の入力文列生成方法の情報の入出力を示す図である。

【図１７】本発明の実施例７における証明付暗号化装置の詳細を示す図である。

【図１８】本発明の実施例７における暗号化検証装置の詳細を示す図である。

【図１９】本発明の実施例６と実施例７における証明付公開鍵列生成装置の詳細を示す図である。

【図２０】本発明の実施例６と実施例７における公開鍵列検証装置の詳細を示す図である。

【符号の説明】

１００　入力暗号文列
１０１　暗号シャッフル
１０２　出力暗号文列
１０３　疑似出力暗号文列
１０４　挑戦値生成関数
１０５　挑戦値
１０６　応答
２００　置換
３００　入力文列
３０１　入力暗号文列
３０３　公開鍵
３０３　再暗号シャッフル情報
３０４　再暗号シャッフル行列
３０５　再暗号秘密乱数
３０６　情報隠蔽
３０７　シャッフル行列
３０８　元係数
３０９　準元係数
３１０　係数基底
３１１　その他
３１２　証明付き再暗号シャッフル行列装置
３１３　出力暗号文列
３１４　暗号シャッフル証明文
３１５　変換情報保有コミットメント

３１６　変換条件コミットメント
３１７　応答
３１８　準応答
３１９　再暗号シャッフル検証装置
４００、６００、８００、１０００　入力文列
４０、６０１、８０１、１００６　再暗号シャッフル情報
４０２、６０２、８０２、１００１　再暗号シャッフル行列
４０３、６０３、８０３、８０５、１００２、１００５　元係数
４０４、６０４、６０５、８０６、１００３　係数基底
４０５、６０２、８０８、１００９　再暗号シャッフル行列作用
４０６、６０３、８０９、１０１０　出力文列
４０７、６０４、８１０、１０１１　出力暗号文列
４０８、６０５、８１１、１０１２　保有コミット
４０９、６０６、８１２、８１６、１０１３、１０２２　恒等式係数計算
４０Ａ、６１４、８２３、１０１７　コミットメント
４１０、６０７、８１３、８１７、１０１４、１０２３　恒等式係数
４１１、６０８、８１４、８１８、１０１５、１０２４　隠蔽処理
４１２、６１３、８１５、８２２、１０１６、１０２８　条件コミット
４１３、６１５、８２４、１０３０、１０３４　挑戦値生成
４１４、６１６、８２５、１０３１、１０３５　挑戦値
４１５、６１７、８２６、１０３２、１０３６　応答生成
４１６、６１８、８２７、１０３３、１０３７　応答
４１７、８１０、１０１１　出力暗号文列
４１８、６２２、８３１、１０４０　再暗号シャッフル証明文
４１９、６２３、８３２、１０４２　変数情報保有コミットメント生成処理
４２０、６２５、８３４、１０４３　変換条件コミットメント生成処理
４２１、６２４、８３５、１０４６、１０４７　応答生成処理
５００　挑戦値生成関数
５０１　挑戦値
５０２、７０６、９０２、１１０３、１１０５　保有検証
５０３　条件検証
５０４、７０９、９０６、１１１０　検証結果
５０５、７１０　　変換情報保有検証処理
５０６、７１１　変換条件検証処理
６１０、１０２６　準元係数隠蔽処理
６１２、８２１、１０２７　準応答コミット
６１９、８２８　準応答生成
６２０、８２９　準応答
６０Ａ、８０７、１００７　準元係数
７０４、９００、１１００　挑戦値生成
７０５、９０１、１１０１　挑戦値
７０７、９０３、９０４、１１０２　恒等式検証
７０８、９０５、１１０７　準応答検証
８２０　準元係数隠蔽処理
８２１　準応答コミット
１００４　情報隠蔽因子
１０１８　選択処理
１０１９　入力文列
１０２０　保有生成
１０２１　保有コミット
１１１１、１１１４　条件検証処理
１１１２、１１１３　保有検証処理
１２００、１４００　基底生成関数
１２０１　入力ベクトル
１３００、１５０１、１９０１　公開鍵列情報
１３０１、１５０２、１９０２　秘密鍵
１３０２、１５０３、１９０３　疑似秘密鍵
１３０４、１５０４　証明付き公開鍵列装置
１３０５、１３０６、１５０５、１５０６　分散公開鍵列対
１３０９　前処理装置
１３０８　検証結果
１４０１、１４０３　公開鍵列基底
１４０２　前処理
１４０４　公開鍵列
１５００　共通初期値
１５０７　公開鍵列検証装置
１５０８　検証結果
１５０Ａ　共通初期値
１６００　共通公開鍵
１６０１　個別公開鍵
１６０２　平文
１６０３　暗号化技術
１６０４　秘密乱数
１６０５　疑似秘密乱数
１６０５　証明付き暗号化装置
１６０７　入力暗号文
１６０８　暗号化証明文
１６０９　暗号化検証装置
１６１０、１８１０　検証結果
１７００、１７０１、１７０２　べき剰余演算
１７０３　暗号文底
１７０４　コミットメント
１７０６、１９１１　挑戦値生成
１７０７、１９１２　挑戦値

１７０８、１９１３　応答生成
１７０９、１９１４　応答
１７１０、１９０４　基底生成関数
１７１１　公開鍵基底
１７１２　平文
１７１３　暗号文対
１８００、２０００　挑戦値生成関数
１８０１、２００３　挑戦値
１８０２　暗号化検証
１９００　　初期値
１９０４、２０００　基底生成関数
１９０５、２００２　公開鍵列底
１９０６　公開鍵列対生成
１９０７　公開鍵列対
１９０８　疑似公開鍵列対生成
１９１５　公開鍵列証明文
１９１７　公開鍵列対
２００４　公開鍵列検証